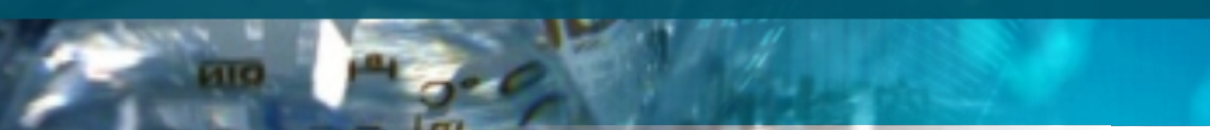
Penerbit
TOHAR MEDIA

Editor :
Wahyuddin S
Nurdjanah Hamid

PENGANTAR DATA SCIENCE

M.Syauqi Haris, Qomarotun Nurlaila, Ratnadewi, John Friadi,
Satya Arisena Hendrawan, Dwiny Meidelfi, Rofiq Harun, Defni,
Garcia, A. Ratna Sari Dewi, Aisyah Mutia Dawis, Garcia KN Ginting



PENGANTAR DATA SCIENCE

Penulis

M. Syauqi Haris, Qomarotun Nurlaila, Ratnadewi, John Friadi, Satya Arisena Hendrawan, Dwiny Meidelfi, Rofiq Harun, Defni, Garcia, A. Ratna Sari Dewi, , Aisyah Mutia Dawis, Garcia KN Ginting

Editor

Wahyuddin S
Nurdjanah Hamid

Penerbit

TOHAR MEDIA

PENGANTAR DATA SCIENCE

Penulis :

M. Syauqi Haris, Qomarotun Nurlaila, Ratnadewi, John Friadi, Satya Arisena Hendrawan, Dwiny Meidelfi, Rofiq Harun, Defni, Garcia, A. Ratna Sari Dewi, , Aisyah Mutia Dawis, Garcia KN Ginting

Editor : Wahyuddin S, Nurdjanah Hamid

ISBN : 978-623-8148-23-3

Desain Sampul dan Tata Letak

Ai Siti Khairunisa

Penerbit

CV. Tohar Media

Anggota IKAPI No. 022/SSL/2019

Redaksi :

JL. Rappocini Raya Lr 11 No 13 Makassar

JL. Hamzah dg. Tompo. Perumahan Nayla Regency Blok D No.25 Gowa

Telp. 0852-9999-3635/0852-4352-7215

Email : toharmedia@yahoo.com

Website : https://toharmedia.co.id

Cetakan Pertama Februari 2023

Hak Cipta dilindungi undang-undang. Dilarang memperbanyak sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronik maupun mekanik termasuk memfotocopy, merekam atau dengan menggunakan sistem penyimpanan lainnya, tanpa izin tertulis dari penerbit.

Undang-undang Nomor 19 Tahun 2002 Tentang Hak Cipta

1. Barang siapa dengan sengaja dan tanpa hak mengumumkan atau memperbanyak suatu ciptaan atau memberi izin untuk itu, dipidana dengan pidana penjara paling lama 7 (Tujuh) tahun dan/atau denda paling banyak **Rp. 5.000.000.000,00 (Lima Miliar Rupiah)**
2. Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta atau hak terkait sebagaimana dimaksud pada ayat 1, dipidana paling lama **5 (lima tahun)** dan/atau denda paling banyak **Rp. 500.000.000,00 (Lima Ratus Juta Rupiah**

Kata Pengantar

Pujian dan limpahan syukur selalu kami panjatkan kehadirat Allah SWT yang telah memberikan bermacam nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan buku dengan judul "**PENGANTAR DATA SCIENCE**" dengan tepat waktu tanpa ada kendala yang berarti. Adapun tujuan dari penulisan ini adalah untuk memudahkan orang banyak mengetahui apa manfaat dari rumput laut dan pengelolaan hingga sampai ke tahap konsumsi.

Kesuksesan dalam penyusunan buku ini tentunya bukan semata atas usaha dari penulis saja. Ada banyak pihak yang turut membantu dalam tercapainya kesuksesan yang sekarang ini dapat dirasakan. Baik dari pihak yang membantu dalam bentuk dukungan moril maupun materil. Penulis buku ini pun dari berbagai universitas yang ada di Indonesia, dengan itu ada banyak pemahaman yang bisa kita ambil dari buku ini. Oleh karena itu sang penulis mengucapkan terima kasih yang sebesar-besarnya kepada seluruh pihak yang turut berpartisipasi dalam tercapainya kesuksesan dari buku ini.

Buku yang ada dihadapan para pembaca sekalian ini pasti memiliki sangat banyak kekurangan dan masih jauh dari kata sempurna. Sehingga penulis banyak berharap kepada para pembaca sekalian untuk memberikan kritik dan sarannya agar buku ini dapat menjadi buku yang lebih sempurna dan lengkap.

Bandung, 14 Desember 2022

Penulis

DAFTAR ISI

PENGANTAR DATA SCIENCE

Penulis

M. Syauqi Haris, Qomarotun Nurlaila, Ratnadewi, John Friadi, Satya Arisena Hendrawan, Dwiny Meidelfi, Rofiq Harun, Defni, Garcia, A. Ratna Sari Dewi, , Aisyah Mutia Dawis, Garcia KN Ginting

Bab 1

Konsep Data Science

1.1. Data Science Saat Ini

Saat ini kita hidup di dunia yang penuh dengan data. Semua situs web melacak setiap klik kita sebagai pengguna. Ponsel cerdas (smartphone) mencatat lokasi dan kecepatan kita setiap detik setiap hari. Alat IoT seperti jam tangan pintar (*smartwatch*) yang kita kenakan selalu mencatat detak jantung, kebiasaan gerakan, pola makan, dan pola tidur kita. Mobil dengan sistem komputer cerdas (*smart-car*) mengumpulkan kebiasaan mengemudi, rumah dengan sistem cerdas (*smart-home*) mengumpulkan kebiasaan hidup, dan sistem pemasar pintar (*smart marketing* system) mengumpulkan kebiasaan para pembeli dan menjadikannya sebagai sistem rekomendasi. Internet itu sendiri mewakili grafik pengetahuan yang sangat besar yang salah satunya berisi ensiklopedia referensi silang yang sangat besar; basis data spesifik tentang film, musik, olahraga, permainan, kuliner, bahkan sampai dengan tempat nongkrong kita; dan banyak sekali data statistik yang dipublikasikan, misalnya data BPS oleh pemerintah.

Terdapat jargon yang mengatakan seorang *data scientist* adalah seseorang yang mengetahui lebih banyak statistik daripada seorang ilmuwan komputer dan lebih banyak ilmu komputer daripada seorang ahli statistik. Meskipun mungkin definisi itu tidak tepat seratus persen, namun faktanya beberapa *data scientist* adalah seorang ahli statistik, sementara yang lain

adalah ahli teknik informatika seperti pengembang perangkat lunak misalnya. *Data Science* adalah ilmu yang bisa dipelajari siapa saja saat ini, mungkin Anda akan menjumpai beberapa ahli pembelajaran mesin (*machine* learning) bahkan tidak memiliki gelar akademik yang istimewa meskipun beberapa diantaranya adalah seorang doktor dengan catatan publikasi yang mengesankan, sementara yang lain bahkan mungkin tidak pernah membaca artikel publikasi akademis

Meskipun demikian, melalui buku ini diharapkan bisa memberikan panduan bagi Anda yang ingin mempelajari *data science*. Seorang *data scientist* adalah seseorang yang mengekstraksi wawasan dari data yang berantakan. Dunia saat ini penuh dengan orang yang mencoba mengubah data menjadi wawasan. Facebook meminta Anda untuk mencantumkan kota asal dan lokasi Anda saat ini, seolah-olah untuk memudahkan teman Anda menemukan dan terhubung dengan Anda. Tapi sebenarnya data tersebut akan digunakan sebagai dasar untuk menganalisis lokasi-lokasi dan mengidentifikasi pola migrasi global dan di mana basis penggemar dari berbagai tren produk dan jasa. Sebagai perusahaan distributor besar, perusahaan akan melacak pembelian dan interaksi konsumen, baik online maupun di toko. Dan itu menggunakan data untuk memprediksi model pelanggan dengan kategori tertentu untuk memasarkan pembelian yang berhubungan dengan kebiasaan secara lebih baik kepada konsumen tersebut.

Banyak orang membayangkan bahwa *data science* sebagian besar adalah *machine learning* dan bahwa *data scientist* sebagian besar membangun dan melatih serta menyesuaikan model pembelajaran mesin sepanjang hari. Faktanya, *data science* sebagian besar adalah mengubah masalah bisnis menjadi masalah data kemudian mengumpulkan data-data yang berhubungan dengan masalah tersebut, memahami data, membersihkan data, dan memformat data agar bisa diproses menggunakan pengolah data *machine learning.* Setelah itu proses

machine learning hampir otomatis dilakukan oleh perangkat-perangkat pemrograman siap pakai. Meski begitu, *machine learning* tetaplah menarik dan penting yang harus Anda ketahui untuk melakukan proses *data science*.

Seorang ahli *data science* atau disebut sebagai *data scientist* dianggap sebagai pekerjaan paling menarik di abad ke-21. Pada 2015, pemerintah Amerika Serikat mengangkat seorang Kepala *Data Scientist* selama masa jabatan Presiden Barack Obama. Tapi apa yang dilakukan seorang *Data Scientist*, dan kualifikasi apa yang dibutuhkan?

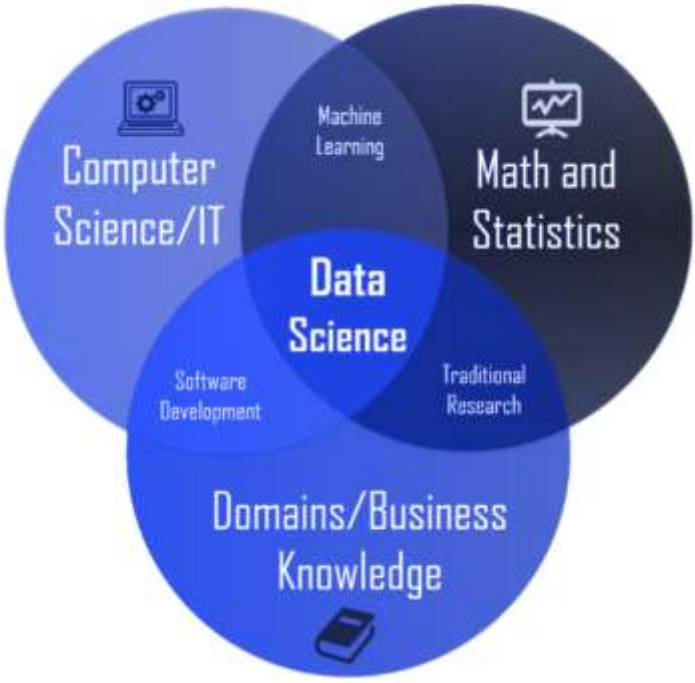
1.2. Pengertian Data Science

Data Science didefinisikan berdasarkan pendekatan multi disiplin keilmuan sebagai gabungan dari statistika, informatika terapan, komputasi, komunikasi, sosiologi, dan manajemen. Disiplin-disiplin ilmu tersebut dianggap mewakili definisi *data science* yang terdiri dari 3 (tiga) komponen utama yaitu data, lingkungan, dan pemikiran. Dari sudut pandang ahli statistik, *data science* dapat dilihat sebagai kombinasi dari statistik dan ilmu komputer. Statistik adalah salah satu disiplin ilmu terpenting yang menyediakan alat dan metode untuk mendapatkan wawasan yang lebih dalam dari data. Sementara itu, dari sudut pandang Teknik, *data science* terdiri dari 4 (empat) bidang inti yaitu rekayasa data, analitik data, prediksi data, dan pembelajaran mesin. Rekayasa data mencakup proses dan metode untuk menyimpan, mengakses, dan ketertelusuran data, sementara analitik data berfokus pada analisis data. Prediksi topik dan situasi berdasarkan pengalaman adalah subjek prediksi data. Pembelajaran mesin dipandang sebagai area penampang untuk ketiga area ini

Inti dari data yaitu sains adalah merupakan data itu sendiri yang diinterpretasikan. Ada banyak sekali informasi mentah yang dialirkan dan kemudian disimpan di gudang data (*data warehaouse*). Ada banyak hal yang bisa dipelajari dengan

cara menambangnya (*data mining*). Bahkan ada kemampuan canggih yang dapat dibangun darinya. Dari sini kita bisa ambil kesimpulan bahwa *data science* pada dasarnya adalah bagaimana menggunakan data dengan cara yang kreatif untuk menambah nilai bisnis. Secara sederhana bisa digambarkan aliran datanya dimulai dari sebuah gudang data atau *data warehouse* yang kemudian dilakukan proses Penemuan Data, wawasan dan analisis kuantitatif untuk membantu pengambilan keputusan bisnis strategis, yang kemudian dilanjutkan dengan proses Pengembangan Produk Data, solusi algoritmik dalam produksi dan pengoperasian. Sehingga pada akhirnya diperoleh sebuah nilai bisnis yang dapat dikomersialisasi

Data Science adalah istilah umum untuk pembelajaran mesin (*machine learning*), pembelajaran mendalam (*deep learning*), visi komputer (*computer vision*), dan pemrosesan bahasa alami (*natural language processing*) sebagaimana dilustrasikan pada Gambar 1.1 Saat kita memiliki data dalam jumlah besar, *data scientist* akan mengubahnya menjadi konten yang bermanfaat, yang berarti mereka akan melakukan semua jenis pemrosesan menggunakan algoritme dan rumus matematika. Seorang *data scientist* akan melihat data dari berbagai perspektif, terkadang mengungkap data yang sebelumnya tidak diketahui. Seringkali setelah melakukan beberapa operasi mereka akan menemukan makna lain di dalamnya.



Gambar 1.1 Posisi Keilmuan *Data Science*

1.3. Bagaimana Data Science Bekerja

Aspek utama *data science* adalah menemukan sesuatu yang baru dan berharga dari kumpulan data. Orang-orang menjelajah pada tingkat terperinci untuk memahami dan menggali kesimpulan, perilaku, dan tren yang kompleks. Semua dilakukan untuk mengungkap informasi tersembunyi yang mungkin dapat membantu perusahaan membuat pilihan yang lebih cerdas untuk bisnis mereka. Sebagai contoh:

- *Data Mining* di Netflix digunakan untuk mencari pola menonton film untuk lebih memahami minat pengguna dan membuat keputusan tentang seri Netflix yang harus mereka hasilkan.
- Divisi Marketing sebuah perusahaan mencoba menemukan segmen pelanggan utama dalam basis pelanggannya dan

perilaku belanja mereka, yang membantu mereka memandu pengiriman pesan ke kelompok pasar lain.

- Perusahaan Proctor & Gamble (P&G) mencari model deret waktu (*time series* model) untuk membantu mereka memahami permintaan di masa mendatang dan merencanakan tingkat produksi.

Jadi bagaimana seorang *data scientist* menggali semua informasi ini? Ini dimulai dengan eksplorasi data. Ketika seorang *data scientist* diberi pertanyaan yang menantang, mereka menjadi seorang detektif. Mereka akan mulai menyelidiki petunjuk, lalu mencoba memahami karakteristik atau pola dalam data. Ini berarti mereka membutuhkan banyak kreativitas analitis. Kemudian seorang *data scientist* dapat menggunakan teknik kuantitatif untuk menyelam lebih dalam, seperti eksperimen kontrol sintetik, peramalan deret waktu, segmentasi, dan model inferensial. Tujuan dari ini adalah untuk menggunakan data untuk menyatukan pemahaman informasi yang lebih baik. Penggunaan wawasan berbasis data inilah yang membantu memberikan panduan strategis. Ini berarti bahwa seorang ilmuwan data bekerja sangat mirip dengan konsultan, membimbing bisnis tentang bagaimana mereka harus menanggapi temuan mereka.

Salah satu contoh klasik produk data adalah mesin yang mengambil data pengguna, lalu membuat rekomendasi yang dipersonalisasi berdasarkan data tersebut. Berikut adalah beberapa contoh produk data:

- Mesin rekomendasi yang digunakan Amazon menyarankan item baru kepada penggunanya, yang ditentukan oleh algoritme mereka. Spotify merekomendasikan musik baru. Netflix merekomendasikan film baru.
- Filter spam di Gmail adalah produk data. Ini adalah algoritme di belakang layar yang memproses surat masuk dan memutuskan apakah itu sampah atau bukan.

- *Computer Vision* yang digunakan untuk mobil *self-driving* juga merupakan produk data. Algoritme pembelajaran mesin dapat mengenali pejalan kaki, lampu lalu lintas, mobil lain, dan sebagainya.

Produk data (*data* product) bekerja secara berbeda dari wawasan data (*data insight*). Wawasan data membantu memberikan beberapa saran untuk membantu eksekutif bisnis membuat keputusan yang lebih cerdas. Produk data adalah fungsionalitas teknis yang mencakup algoritme, dan dirancang untuk berfungsi dalam aplikasi utama. *Data scientist* memainkan salah satu peran paling sentral dalam menghasilkan produk data. Ini berarti bahwa mereka harus membuat algoritme dan menguji, menyempurnakan, dan menerapkannya secara teknis ke dalam sistem produksi. *Data scientist* juga bekerja sebagai pengembang teknis dengan menciptakan aset yang menjadi daya ungkit dalam skala luas

1.4. Apa Yang Harus Dipelajari

Saat berhadapan dengan *data science*, ada tiga bidang keterampilan utama yang digabungkan menjadi satu, yaitu:

1. Keahlian matematika;
2. Literasi Teknologi/Keterampilan Pemrograman;
3. Naluri bisnis/strategi.

Data science adalah tempat ketiganya bersatu. Inti dari menambang data dan membuat produk data adalah melihat data secara kuantitatif. Terdapat korelasi, tekstur, dan dimensi pada data yang dilihat secara matematis. Menemukan solusi melalui data menjadi tantangan yang menarik secara teknik kuantitatif dan heuristik. Untuk menemukan solusi dalam banyak masalah melibatkan pembuatan model analitik yang didasarkan pada matematika murni. Memahami mekanisme di bawah model tersebut sangat penting untuk kesuksesan dalam menghasilkan produk data.

Ada juga kesalahpahaman besar bahwa *data science* hanya berurusan dengan statistik. Memang benar bahwa statistik memainkan peran penting, namun itu bukan satu-satunya matematika yang digunakan. Ada dua cabang utama statistik: statistik Bayesian dan klasik. Ketika orang mulai berbicara tentang statistik, mereka paling sering berbicara tentang statistik klasik, tetapi memahami keduanya sangat membantu. Saat mempelajari lebih banyak algoritma pembelajaran mesin dan teknik inferensial, kita akan sangat bergantung pada aljabar linier. Salah satu contohnya adalah cara populer untuk menemukan karakteristik tersembunyi dalam kumpulan data adalah dengan *Slowly Changing Direction* (SCD), yang didasarkan pada matematika matriks dan tidak banyak berhubungan dengan statistik klasik. Secara keseluruhan, sangat membantu bagi seorang ilmuwan data untuk memiliki pemahaman matematika yang cukup baik di semua bidang.

Mengapa keterampilan pemrograman itu penting? Terutama karena seorang *data scientist* akan menggunakan teknologi untuk membantu mereka mengumpulkan data dalam jumlah besar, dan kemudian bekerja dengan algoritme yang kompleks untuk memahaminya. Ini akan membutuhkan alat yang cenderung lebih canggih daripada *spreadsheet*. Ilmuwan data harus memahami cara membuat kode, membuat prototipe solusi cepat, dan mengintegrasikan sistem data yang kompleks. Beberapa bahasa paling umum yang terkait dengan ilmu data adalah SAS, R, Python, dan SQL. Beberapa yang kurang umum digunakan adalah Julia, Java, dan Scala. Tapi itu bukan hanya memiliki pemahaman yang baik tentang dasar-dasar bahasa ini. *Data scientist* dengan pemahaman pemrograman yang baik dapat secara kreatif mengatasi berbagai jenis tantangan sehingga mereka dapat membuat kodenya berfungsi.

Penting juga bahwa seorang *data scientist* adalah seorang konsultan bisnis taktis. Karena *data scientist* bekerja erat dengan data, mereka dapat mempelajari hal-hal dari data yang tidak dapat dipelajari orang lain. Ini membuat mereka bertanggung jawab untuk menerjemahkan pengamatan mereka menjadi pengetahuan bersama dan berbagi strategi mereka tentang bagaimana menurut mereka masalah harus dipecahkan. Seorang ilmuwan data harus dapat berbagi cerita yang jelas. Mereka seharusnya tidak membuang data begitu saja. Perlu disajikan dalam diskusi yang kohesif tentang suatu masalah dan solusinya yang menggunakan wawasan data sebagai basisnya. Memiliki ketajaman bisnis memainkan peran yang sama pentingnya dengan memiliki ketajaman pada algoritme dan teknologi. Harus ada kecocokan yang jelas antara tujuan bisnis dan proyek ilmu data. Pada akhirnya, nilainya tidak akan datang dari teknologi, data, dan matematika. Itu akan datang dari memanfaatkan semua informasi ini menjadi hasil yang berharga bagi bisnis

Tugas seorang *data scientist* bisa sangat beragam. Untuk menggambarkan pengetahuan luas yang dibutuhkan *data scientist* untuk melakukan pekerjaan mereka, daftar kualifikasi dibuat dan ditugaskan ke berbagai bidang studi. Representasi kualifikasi dalam bidang studi yang ditugaskan ditunjukkan pada Tabel 1.1.

Tabel 1.1. Kualifikasi Data Scientist

Business / Product Development	***Machine Learning / Big Data***	***Mathematics / Operation Research***	***Programming / System Administration***	***Statistics and Visualization***
• *Business* • *Produc*	• *Big Data and Distribut*	• *Algorithms* • *Bayesian*	• *Back-end programming*	• *Statistics* • *Surveys*

t Development	*ed Data* • *Machine Learning* • *Structured data* • *Unstructured data*	*statis-* • *tics and Monte* • *Carlo methods* • *Graphical models* • *Mathematics* • *Optimization* • *Simulation*	• *Front-end programming* • *System administration*	*and* • *marketing* • *Visualization*

Bab 2

Konsep Statistika Untuk Data Science

2.1. Pendahuluan

Data analitycs dan data science mencakup berbagai bidang ilmu diantaranya kecerdasan dasar, ilmu komputer, pembelajaran mesin, statistik, matematika. Data analytics dilakukan dengan cara menganalisa data dari berbagai sumber, dengan berbagai ukuran dan jenis yang berbeda untuk mendapatkan suatu kesimpulan. Kesimpulan digunakan perusahaan untuk membuat keputusan bisnis yang lebih akurat dan efektif untuk kemajuan bisnis.

Data science adalah penggabungan statistik, ilmu komputer dan ilmu matematika untuk memperlancar proses analisa data. Membuat suatu sistem yang dapat digunakan untuk proses analisa data, system didukung oleh kecerdasan buatan (AI) dan machine learning dengan menerapkan suatu algoritma.

Data science minimal terdiri dari komponen berikut:

1. Statistik. Terkait dengan cara pengumpulan data, analisa dan interpreatsi data serta penyajian data dengan cara matematika.
2. Visualisasi data. Mengubah tampilan data sehingga lebih mudah untuk dipahami. Data bisa diubah dalam bentuk

grafik, chart atau diagram disesuaikan dengan bentuk dan karakteristik data, sehingga lebih mudah untuk dilihat dan dipahami. Visualisasi merupakan cara efektif untuk mengeksplorasi dan mengkomunikasikan hasil data akhir. Melalui visualisasi akan memudahkan dalam mengambil keputusan berdasarkan data dan dapat menambah wawasan bisnis serta menentukan rencana strategi bisnis yang sesuai.

3. Machine learning. Merupakan komponen yang paing krusial dan penting, karena akan menentukan keakuratan analisa data untuk memprediksi tingkah laku dan minat pelanggan. Machine learning bisa mengelola dan mengoperasikan data dalam jumlah besar, dimana proses pengambilan keputusannya berpusat pada data. Pada machine learning, metode analisis data dengan membuat model analistik yang otomatis. Dengan machine learning, bisa dilakukan prediksi data, bisa mendeteksi kemungkinan adanya resiko dan penipuan. Pada machine learning juga bisa dibuat system penganalan suara dan wajah serta penyaringan spam

Perbedaan data analytics dan data science terletak pada cakupan data yang dikerjakan dan perlakuan terhadap data tersebut.. Data analytics merupakan salah satu tahapan pengelolaan data science. Data science diperlukan sebelum proses data analytics, dimana hasil data science akan menentukan keakuratan hasil analisa. Dat science berperan untuk merancang dan membangun proses baru, sehingga dihasilkan pemodelan data dengan menggunakan algoritma. Data analytics dilaksanakan setelah proses data sience, yaitu untuk memeriksa data dalam jumlah besar, sehingga diketahui trend dan kesimpulan. Kesimpulan terebut digunakan untuk perusahaan menyusun strategi yang lebih baik untuk kemajuan bisnis.

Data science melibatkan data dan sains (ilmu). Ilmu digunakan untuk memproses data. Data science terjadi ketika kita ingin mendapatkan informasi berkaitan dengan data yang tersedia. Penekanannya lebih ke data bukan ke ilmu yang digunakan untuk menganalisa. Ketika kita memiliki data dan kita memiliki rasa keingin tahuan yang besar terhadap isi kandungan data tersebut yang bermanfaat untuk kita atau perusahaan tempat kita bekerja. Maka untuk menjawabnya, kita pelajari data yang ada dengan melakukan eksplorasi dan memanipulasinya, menganalisa dengan menggunakan teknologi dan ilmu yang bisa membantu kita mendapatkan jawaban.

Data science mempelajari data terutama data kuantitatif. Pada data science dilakukan proses penggalian data sehingga diproduksi pengetahuan data. Untuk menghasilkannya dilakukan melalui tahapan antara lain desain, pengumpulan dan analisis data. Pada data science data disajikan lebih bermakna dan logis.

Di era industry 4.0, mayoritas perusahaan membutuhkan data science, sehingga mampu bersaing. Berbagai alat, mesin atau sistem dirancang menjadi pintar sehingga bisa berpikir dan memutuskan sendiri. Alat, mesin atau system dilengkapi dengan kemampuan mengambil dan menganalisis data, atau mengambil informasi dari alat, mesin atau system lain. Informasi tersebut merupakan dasar untuk melakukan aksi.

Kebutuhan dalam Data Science mencakup:

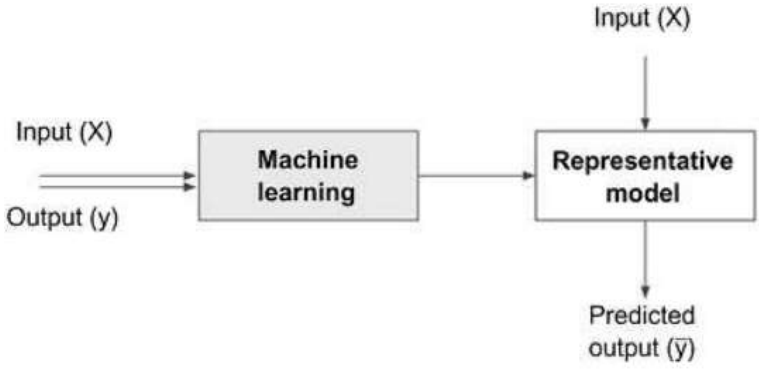
1. Pembelajaran mesin merupakan tulang punggung dari data science.
2. Pemodelan, model matematika memungkinkan untuk membuat perhitungan dan prediksi dengan cepat berdasarkan apa saja yang dapat diketahui terkait dengan data. Pemodelan merupakan bagian dari pembelajaran mesin dan melibatkan identifikasi algoritma mana yang

paling cocok (sesuai) untuk memecahkan masalah dan bagaimana melatih model-model tersebut.

3. Statistik merupakan inti dari data science (ilmu data). Pemahaman yang kuat tentang statistik membantu dalam proses mengekstrak intelijen dan memperoleh hasil yang bermakna.
4. Pemorgraman, beberapa level pemrograman diperlukan.
5. Database, seorang ilmuwan perlu memahami cara kerja database, cara mengekstrak data dan cara mengelolanya.

2.2. Model Data Science

Data science dapat menemukan pola sebuah model representatif dengan menggunakan pembelajaran mesin. Model representatif dapat memberikan gambaran hubungan antar variabel. Data representatif dibentuk dari data pengamatan, yang diharapkan bisa digunakan untuk memprediksi output berdasarkan input serta dapat digunakan untuk memahami hubungan antara output dan input.



Gambar 2.1. Model data science

2.3. Statistika Dalam Data Science

Statistika merupakan ilmu untuk mengumpulkan, menyusun, menginterpretasikan, menganalisis dan menyajikan data sehingga pengambilan keputusan dapat dilakukan secara

efektif. Metode yang tepat untuk pengolahan data-data angka adalah statistika. Untuk memahami data science diperlukan statistika. Statistika merupakan bidang ilmu yang fokus pada data angka. Dalam statistika, data mentah dikumpulkan untuk mempelajari suatu masalah. Statistik sudah digunakan untuk menganalisis data sebelum data science ditemukan. Sebagai contoh untuk mendapatkan sebaran data, rangkuman data, pengujian hipotesis, membuat sampel data dan melakukan analisis multivariate. Statistika digunakan untuk

1. Memahami data, disaat mempelajari dan mengeksplorasi data.
2. Membersihkan data ketika ada data yang tidak konsisten atau salah.
3. Mentransformasi data, mengubah suatu nilai data
4. Menguji cakupan berupa model, mengukur tingkat kebenaran model atau membandingkan model-model yang ada untuk dipilih yang sesuai dan terbaik.

Bagi seorang peneliti, statistika berguna untuk:

1. Menyusun, menyederhanakan atau meringkas data karena data suatu penelitian mencakup banyak informasi.
2. Merancang (merencanakan) kegiatan eksperimen untuk mendapatkan informasi dengan biaya terkecil, ini terkait dengan metodologi dan pengambilan kesimpulan secara statistika.
3. Menetapkan metode terbaik dalam pengambilan kesimpulan sesuai dengan pengambilan sampel. Penarikan kesimpulan bisa digunakan untuk melakukan prediksi atau membuat suatu keputusan.
4. Mengukur atau mengevaluasi baik tidaknya suatu pengambilan kesimpulan.

Tabel 2.1 Perbandingan Statistika Dan Data Science

Aspek	Statistika	Data science
Konsep	Statistika adalah ilmu data.	Berdasarkan teknologi komputasi ilmiah.
	Digunakan untuk mengukur atau memperkirakan atribut.	Meliputi pembelajaran mesin, proses analitik lainnya, model bisnis.
	Menerapkan fungsi statistik atau algoritma pada kumpulan data untuk menentukan nilai yang sesuai untuk masalah yang sedang dipelajari.	Menggunakan matematika dan statistik tingkat lanjut untuk mendapatkan informasi baru dari data besar. Disiplin yang luas yang melibatkan pemrograman, pemahaman tentang model bisnis, tren, dan sebagainya.
Pendekatan	Penggunaan rumus, model, dan konsep matematika.	Menerapkan metode ilmiah dalam pemecahan masalah menggunakan data acak.
	Analisis data acak.	Mengidentifikasi persyaratan data untuk masalah tertentu.

Memperkirakan nilai untuk atribut data yang berbeda.	mengidentifikasi teknik untuk mendapatkan hasil yang diinginkan.
Untuk menentukan perilaku berdasarkan data.	Memberikan nilai kepada organisasi menggunakan data.

Sumber: https://www.educba.com/data-science-vs-statistics/

2.4. Jenis data Statistika

A. Statistika deskriptif

Membahas tentang pengumpulan dan penyederhanaan angka-angka pengamatan, serta melakukan pengukuran penyebaran dan pemusatan, sehingga didapatkan rangkuman data yang lebih mudah dipahami, menarik dan berguna. Penyajian data berbentuk diagram, grafik atau menyajikan ukuran penyebaran dan pemusatan. Kegunaan dari statistika deskriptif:

1. Data tersaji dengan rapi dan ringkas serta bisa menunjukkan inti informasi dari kumpulan data.
2. Memungkinkan peneliti menggambarkan datanya dengan grafik atau angka.
3. Memungkinkan peneliti meneliti hubungan anata dua variabel dengan terlebih dahulu mengukur ua variabel dari setiap responden.
4. Sebagai tahap persiapan dalam analisis data, peranannya sangat penting

B. Statistika Inferensia

Membahas cara menganalisis data dan mengambil keputusan, yang berkaitan dengan perkiraan parameter dan pengujian hipotesis. Metodenya berkaitan dengan analisis sampel sampai ke peramalan atau pengambilan keputusan terhadap populasi.

Diadakan pendugaan parameter, pembuatan dan pengujian hipotesis dan pengambilan keputusan yang berlaku umum. Karakteristik dari statistika inferensia antara lain:

1. Pengamatan secara acak, pengamatan atau pengukuran tidak dapat diprediksi.
2. Teknik penarikan sampel, dilakukan secara acak dari data yang tersedia.
3. Data dalam bentuk angka, merupakan hasil pengukuran karakteristik setiap elemen. Terhadap hasil data kualitatif perlu diberikan simbol angka untuk masing-masing kategori sehingga pada akhirnya pengukuranya bisa dilakukan secara kuantitatif.
4. Tujuan umum inferensia. Pengambilan sampel dilakukan secara acak sehingga akan didapatkan informasi dari setiap elemen populasi yang diperhatikan, dengan harapan dapat diambil kesimpulan dari keseluruhan populasi.

C. Statistika Parametrik

Nilai dari parameter populasi dipertimbangkan. Data yang dibutuhkan berskala interval, penetapan teori dan penurunan prosedur berpedoman pada asumsi bahwa bentuk distribusi populasinya adalah normal.

D. Statistik nonparametrik

Nilai dari parameter populasi tidak diperhatikan. Validitasnya tidak tergantung pada model peluang dari populasi. Menyediakan metode analisisi data yang distribusinya tidak normal. Data yang dibutuhkan berskala ordinal dan nominal.

2.5. Pengukuran dan Perbandingan Data

Proses pengukuran adalah proses deduktif, berangkat dari suatu konsep atau ide dan dilanjutkan susun perangkat ukur untuk pengamatan empiris. Proses pengukuran diawali dengan konseptualisasi untuk variabel-variabel dan konsep-

konsep yang akan masuk dalam hipotesis. Konsep-konsep dipilih dan diberikan secara teoritis. Konseptual merupakan batasan dalam tingkat yang abstrak. Batasan harus memiliki makna yang jelas, khusus dan eksplisit. Langkah selanjutnya adalah operasionalisasi untuk menyusun batasan operasional untuk konsep terpilih. Batasan operasional merupakan batasan konsep berbentuk prosedur, instrument maupun metode pengukuran sehingga memungkinkan pengamatan variabel empiris. Ukuran empiris menggambarkan secara nyata ukuran variabel dengan mengacu pada alat indikator, sehingga keberadaaan konsep ditunjukkan dalam kenyataan yang diamati.

Penghubung antara indikator dan konsep penting dalam proses pengukuran, proses pengukuran deduktif melalui tahapan:

1. Tingkat konseptual, untuk merumuskan batasan yang jelas pada suatu konsep.
2. Tingkat operasional, untuk menyusun definisi operasional atau seperangkat indikator untuk konsep tersebut.
3. Tingkat empiris, menerapkan indikator pada dunia nyata.

Untuk menguji hipotesis yang bersifat empiris digunakan penghubung antar kenyataan empiris dan konsep yang abstrak. Pengujian empiris dihubungkan kembali pada hubungan sebab-akibat serta hipotesis konseptual.

Gambar 2.2. Proses pengukuran

Sumber: Modul

Prinsip pengukuran:

1. Prinsip eksklusif, suatu kejadian hanya memiliki satu nilai untuk variabel yang sama. Contoh variabel jenis kelamin, yang memiliki jenis kelamin perempuan pada saat yang sama tidak bisa memiliki jenis kelamin laki-laki.
2. Prinsip ekshaustif, nilai yang tersedia untuk satu variabel harus mencakup nilai dari seluruh kemungkinan kejadian. Contoh variable alat transportasi, harus dapat mencakup seluruh kemungkinan jawaban.

A. Konsep Dasar dalam Proses Pengukuran

A.1. Variabel dan Konstanta

Variabel berubah-ubah dan tidak tetap. Variabel merupakan besaran yang dapat berubah sehingga mempengaruhi peristiwa, kejadian atau hasil penelitian. Variabel memudahkan pemahaman akan masalah, seolah-olah jawaban sudah didapatkan. Variabel bagian penting dari penelitian sains. Konstanta merupakan variabel yang nilainya tetap dan tidak bisa diubah.

Statistika sebagai alat yang membantu untuk menyajikan data dari gejala yang bervariasi dan merumuskan cara pengambilan kesimpulan secara tepat. Data perubahan variabel didapatkan berdasarkan pengamatan terhadap kasus. Contoh dari variabel antar lain status kesehatan masyarakat dan pertumbuhan penduduk. Contoh dari konstanta antara lain nilai pi (3.14159).

A.2. Variabel Kuantitatif dan Variabel Kualitatif

Variabel kuantitatif merupakan variabel bervariasi dalam hal jumlah pada setiap data pengamatan. Contohnya: angka kelahiran, angka buta huruf, kepadatan penduduk, umur, berat badan, tinggi badan. Variabel kualitatif merupakan variabel yang bervariasi dalam jenis, sebagai contoh: agama, pekerjaan,

jenis kelamin, suku bangsa, status perkawinan. Untuk identifikasi pada variabel kualitatif diberikan angka yang tidak bisa dilakukan operasi matematis.

A.3. Variabel Kuantitatif

Variabel kuantitatif dibedakan menjadi variabel :

1. Diskrit, jumlah kategori (nilai) dapat dihitung yaitu berupa bilangan bulat. Contoh: Jumlah anggota keluarga, jumlah siswa dalam suatu kelas, dan lain-lain.
2. Kontinu, hasil pengamatannya adalah salah satu dalam garis interval, sehingga nilainya bisa berupa nilai pecahan maupun genap. Contoh: umur, masa kerja, tinggi badan, berat badan. Umur dan masa kerja datanya berupa bulan dan tahun, tinggi badan dalam meter (1.2 meter), berat badan dalam Kg (5.2 Kg).

Perbedaan variabel diskrit dan kontinu diperlukan karena akan menentukan metode dalam membuat keputusan.

B. Skala Pengukuran

B.1. Skala Nominal

Skala nominal digunakan untuk mengukur variabel kualitatif. Kategori variabel kualitatif diberikan berdasarkan "nama" dan diberikan angka untuk keperluan identifikasi, tetapi tidak menunjukkan suatu besaran. Angka diberikan untuk mempermudah penggambaran karakteristik data dan analisis. Contoh: pendidikan, status perkawinan, pekerjaan.

B.2. Skala Ordinal

Memiliki sifat seperti data nominal, hanya diberikan tambahan informasi dan memiliki karakteristik tambahan sehingga dapat disusun berdasaran suatu urutan. Contohnya: jenjang pendidikan, posisi dalam pekerjaan, golongan PNS, dan lain-lain.

B.3. Skala Interval

Dapat ditentukan bahwa suatu data lebih atau kurang dari data lainnya. Skala interval mencakup seluruh sifat skala nominal dan ordinal. Sifat tambahan dari skala interval adalah bisa menetapkan jarak antar kategori yang tersedia pada alternatif jawaban. Contohnya suhu ruangan, dimana skalanya sudah ditentukan dan tetap, yaitu 1 derajat. Keterbatasan dari skala interval adalah tidak diketahuinya titik awal dari skala pengukuran atau titik nol tidak bisa ditentukan.

B.4. Skala Rasio

Skala rasio memiliki seluruh sifat dari skala nominal, ordinal, dan interval serta ditambah kemampuan membandingkan skala pengukuran yang disusun. Pada skala rasio terdapat nilai nol, bisa menunjukkan jika tidak ada jumlah yang dapat diamati untuk suatu variabel sehingga . memungkinkan untuk membandingkan antar kategori yang tersedia.

Dalam pembuatan skala pengukuran perhatikan hal-hal berikut:

1. Variabel kualitatif: skala nominal.
2. Variabel kuantitatif: skala rasio, interval dan ordinal
3. Skala ordinal: informasi paling sedikit informasi (peringkat kategori suatu skala).
4. Skala interval: Jarak antar kategori dapat ditetapkan, letak titik nol (awal) tidak diketahui.
5. Skala rasio: paling informatif.

C. Perbandingan Data

Bentuk dari perbandingan data antara lain:

1. Rasio, digunakan dalam perbandingan antar dua kelompok data. Perbandingan antar elemen yang ada pada tiap

kelompok (satu elemen tidak bisa menjadi bagian dari dua kelompok data atau lebih).

2. Proporsi, bentuk khusus dari rasio dimana pembaginya merupakan jumlah elemen pada data kelompok A dan kelompok B.
3. Persentase, cara perhitungannya sama dengan proporsi hanya rangenya 1-100, dimana hasilnya dikalikan 100 dan dibelakangnya ditambahkan smbol %.
4. Rates (Tingkat/Angka), membagi jumlah munculnya peristiwa/gejala/kejadian yang dimaksud (misalnya angka kematian bayi) dengan total jumlah yang muncul. Untuk analisa angka rate dikalikan dengan angka tertentu, misalkan dikalikan dengan angka 1000. Sehingga data akan menunjukkan kematian bayi dari 1000 kelahiran bayi.

D. Validitas dan Reliabilitas

Validitas berkaitan dengan keterwakilan variabel. Evaluasi dilakukan dengan memeriksa instrument pengumpulan data yaitu daftar pertanyaan, bagaimana perumusannya dan fokusnya. Selain itu, bisa meminta seseorang ahli untuk mengevaluasi instrument yang digunakan.

Reliabilitas berkaitan dengan keandalan dan konsitensi dari hasil pengukuran variabel yang diteliti. Evaluasi dilakukan dengan prosedur test-retest, yaitu instrument pengukuran diterapkan harus lebih dari sekali, minimal dua kali untuk satu sampel. Instrument handal ketika hasil pengukurannya didapatkan hasil yang sama (tidak berbeda jauh).

2.6. Probabilitas

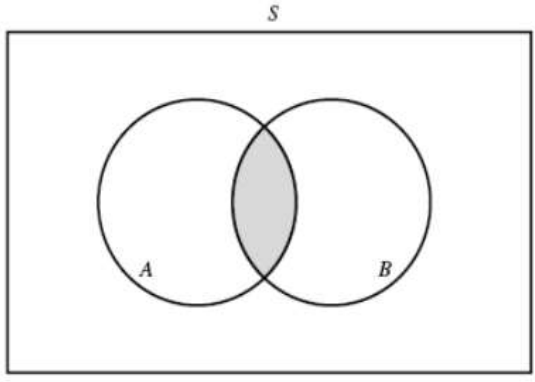
A. Ruang sampel dan Kejadian

Point-point penting:

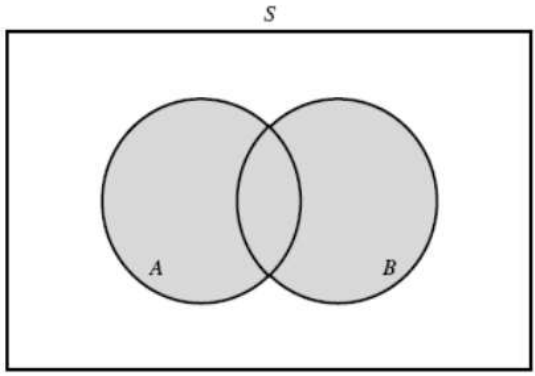
- Percobaan merupakan proses yang menghasilkan pengamatan atau ukuran, bisa sederhana dan rumit serta

tidak harus dilakukan dilaboratorium tetapi bisa dengan hanya dibayangkan.

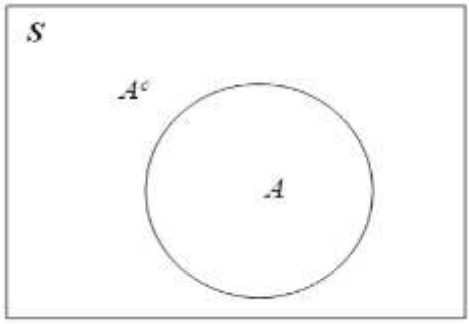
- Kejadian merupakan hasil dari suatu percobaan dan kumpulan dari beberapa kejadian sederhana.
- Kejadian sederhana ketika kejadian tersebut tidak dapat didekomposisikan.
- Dua kejadian A dan kejadian B dikatakan saling asing jika satu kejadian terjadi dimana yang lain tidak mungkin terjadi dan sebaliknya.
- Irisan $A \cap B$ menyatakan bahwa kejadian A dan kejadian B terjadi bersamaan. $A \cap B = \{ x \in S \mid x \in A \text{ dan } x \in B \}$. Diagram Venn ditunjukkan pada gambar 2.3.
- Gabungan $A \cup B$ menyatakan bahwa kejadian A atau kejadian B atau kedua kejadian tersebut terjadi. $A \cup B = \{ x \in S \mid x \in A \text{ atau } x \in B \}$. Diagram Venn ditunjukkan gambar 2.4
- Komplemen kejadian A adalah A^c, semua kejadian sederhana dalam ruang sampel S yang tidak berada dalam A. $A^c = \{ x \in S \mid x \notin A \}$. Gambar 2.5 menunjukkan komplemen A.



Gambar 2.3. Irisan $A \cap B$



Gambar 2.4. Gabungan A ∪ B



Gambar 2.5. Komplemen A (A^c)

B. Analisis kombinatorial

Analisis kombinatorial digunakan untuk menentukan probabilitas, dengan terlebih dahulu menghitung titik sampel. Mencakup aturan membuat pasangan, pemutasi (penyusunan yang memperhatikan urutan) dan kombinasi.

C. Teori Probabilitas

Nilai bobot dari teori probabilitas berkisar antara 0 dan 1, sehingga jumlah bobot dari keseluruhan ruang sampel adalah 1. Postulat dari teori probabilitas :

1. $P(C) \geq 0$, untuk setiap kejadian C.

2. $P(S) = 1$, untuk kejadian pasti S.

3. $P(A_1 \cup A_2 \cup \ldots\ldots) = P(A_1) + P(A_2) + \ldots\ldots$, untuk semua kejadian yang saling asing A_1, A_2, ...,

Berikut beberapa aturan atau ketetapan pada teori probabilitas:

- Jumlah dari probabilitas kejadian sederhana merupakan probabilitas kejadian A.
- Bila suatu percobaan kejadian A mendapatkan hasil m dari kemungkinan hasil M maka P(A) = m/M.
- Jika $B \subset A$ maka $P(B) \leq P(A)$ dan $P(A - B) = P(A) - P(B)$.
- Untuk setiap kejadian A berlaku $0 \leq P(A) \leq 1$.
- $P(\emptyset) = 0$, artinya kejadian mustahil maka probabilitasnya 0.
- Jika A^c adalah komplemen dari kejadian A, maka $P(A^c) = 1 - P(A)$.
- Jika A dan B dua kejadian sembarang, maka $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.
- Dua kejadian A dan B, $P(B) = P(B \cap A) + P(B \cap A^c)$
- Probabilitas bersyarat dari B dimana P(A) > 0) adalah $P(B \mid A) = P(B \cap A) / P(A) = P(A \cap B) / P(A)$.
- $P(A \mid B) = P(A)$ atau $P(B \mid A) = P(B)$. Kejadian A dan B dikatakan saling bebas, jika tidak maka keduanya saling bergantung.
- Hukum multiplikatif probabilitas. Probabilitas dari irisan $P(A \cap B) = P(A) P(B \mid A) = P(B) P(A \mid B)$. $P(A \cap B) = P(A) P(B)$, maka A dan B saling bebas.
- Untuk tiga kejadian sembarang A, B dan C berlaku $P(A \cap B \cap C) = P(A) P(B \mid A) P(C \mid A \cap B)$.

2.7. Penerapan Statistik Untuk Data Science

Data science bertujuan memberikan wawasan yang berarti tentang ketersediaan data dalam jumlah besar. Sehingga perlu untuk melibatkan banyak bidang pekerjaan yang berbeda,

dengan tujuan melakukan perhitungan dan penerjemahan data yang akan disaring. Contoh penerapannya antaralain:

1. **Bidang finansial,** contoh penerapan untuk keperluan fraud detection di bidang finansial yaitu untuk mengkategorikan, mengklasifikasikan, dan mengelompokkan data yang dapat menunjukkan bentuk penipuan. Hal ini untuk menghindari kriminalisasi dan untuk mengenkripsi data yang mampu mendeteksi penipuan dalam bentuk yang tidak terduga.
2. **Bidang kesehatan,** Salah satu contoh penerapan google dapat memetakan wabah flu secara real time dengan melacak data lokasi pada pencarian terkait flu. Google meluncurkan tool yang up-to-date, yaitu Google Flu Trends.
3. **Bidang olahraga professional,** contoh penerapan digunakan untuk merekrut pemain muda dengan potensi bintang dan untuk memprediksi pemain potensial dan membangun tim yang kuat dengan biaya rendah. Penerapan yang lain yaitu untuk mendeteksi dan menyembuhkan penyakit yaitu dengan mengolah data kasus penyakit yang pernah terjadi sebelumnya.
4. **Bidang e-commerce, c**ontoh penerapan untuk memberi berbagai tujuan penetapan harga yang dinamis, sehingga perusahaan e-commerce dapat mengelompokkan pelanggan atau konsumen secara tepat serta agar setiap kelompok pelanggan dapat ditawarkan suatu produk dengan harga yang sesuai dengan kebutuhannya.
5. Bidang Bisnis, contoh penerapan untuk menentukan keputusan bisnis perusahaan. Perusahaan mengolah data pelanggan, pesaing, tren terkini untuk menjadi bahan pertimbangan membuat strategi bisnis. Selain itu diterapkan juga untuk mengetahui anomaly dalam struktur bisnis dan penyebab masalah bisnis.

6. Bidang Teknologi, contoh penerapannya untuk mengidentifikasi objek dan pola dalam gambar untuk *image recognition*, memastikan sensor dalam kendaraan tanpa wak berjalan dengan baik dan lebih aman.

Bab 3

Hierarki Data Clustering

3.1 Pengantar

Clustering adalah pengelompokan data tersimpan, hasil pengamatan, atau kasus ke dalam kelas objek yang sama. *Cluster* adalah kumpulan data tersimpan yang mirip satu sama lain dan berbeda dengan data tersimpan di kelompok lain. *Clustering* berbeda dari klasifikasi karena tidak ada variabel target untuk pengelompokan. Tugas pengelompokan tidak mencoba mengklasifikasikan, memperkirakan, atau memprediksi nilai variabel target. Sebaliknya, algoritma pengelompokan berusaha untuk membagi seluruh data yang ditetapkan menjadi subkelompok atau kelompok yang relatif homogen, dengan kesamaan data dalam *cluster* dimaksimalkan, dan kesamaan dengan catatan di luar *cluster* ini diminimalkan.

Contoh tugas pengelompokan dalam bisnis dan penelitian adalah sebagai berikut:

- Menargetkan pemasaran produk khusus untuk bisnis kapitalisasi kecil yang tidak memiliki anggaran pemasaran besar.
- Untuk tujuan audit akuntansi, untuk mengelompokkan perilaku keuangan ke dalam kategori sehat dan mencurigakan.
- Sebagai alat pengurangan dimensi ketika kumpulan data memiliki ratusan atribut.

- Untuk pengelompokan ekspresi gen, di mana jumlah gen yang sangat besar dapat menunjukkan perilaku serupa.

Clustering sering dilakukan sebagai langkah awal dalam proses *data mining*, dengan cluster yang dihasilkan digunakan sebagai input lebih lanjut ke dalam teknik hilir yang berbeda, seperti jaringan saraf. Karena ukuran yang sangat besar dari banyak basis data saat ini, seringkali berguna untuk menerapkan analisis pengelompokan terlebih dahulu, untuk mengurangi ruang pencarian untuk algoritma hilir.

Analisis klaster menghadapi banyak masalah yang sama yang kita bahas di bab tentang klasifikasi. Misalnya, perlu ditentukan

• bagaimana mengukur kesamaan;

• bagaimana mengkode ulang variabel kategorikal;

• bagaimana menstandardkan atau menormalkan variabel numerik;

• berapa banyak cluster yang diharapkan untuk diungkap.

Untuk mempermudah, dalam buku ini, konsentrasi pada jarak Euclidean antar data tersimpan (D'Agostino & Dardanoni, 2009):

$$d_{Euclidean}(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

dengan $x = x_1, x_2, \ldots, x_m$ dan $y = y_1, y_2, \ldots, y_m$ mewakili nilai atribut m dari dua data tersimpan. Tentu saja, ada banyak metrik lainnya, seperti city-block distance (Melter, 1987):

$$d_{city-block}(x,y) = \sqrt{\sum_i |x_i - y_i|}$$

atau jarak Minkowski, yang mewakili kasus umum dari dua metrik di atas untuk eksponen umum q (Nishom, 2019):

$$d_{Minkowski}(x,y) = \left(\sum_i |x_i - y_i|^q\right)^{1/q}$$

Untuk variabel kategorikal, dapat didefinisikan kembali fungsi "berbeda dari" untuk membandingkan nilai atribut ke-i dari sepasang record:

$$beda(x_i,y_i) = \begin{cases} 0 & jika\ x_i = y_i \\ 1 & untuk\ yang\ lainnya \end{cases}$$

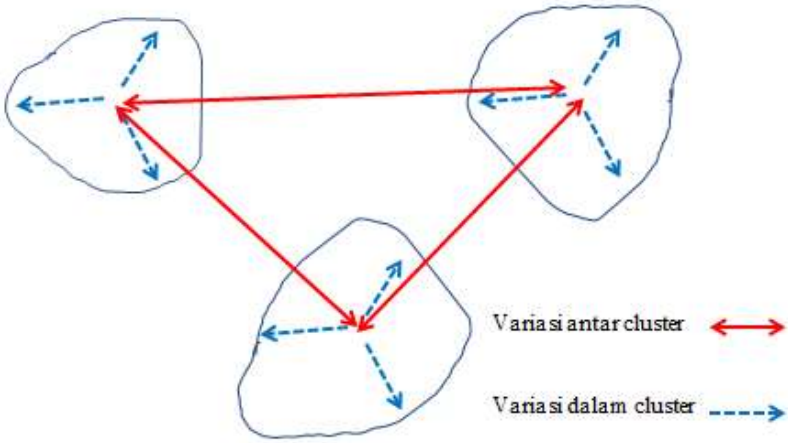
dengan x_i dan y_i adalah nilai kategorikal. Kemudian dapat diganti suku ke-i yang berbeda (x_i,y_i) dalam metrik jarak Euclidean di atas.

Untuk kinerja yang optimal, algoritma *clustering*, seperti halnya algoritma untuk klasifikasi, memerlukan data untuk dinormalisasi sehingga tidak ada variabel atau subset variabel tertentu yang mendominasi analisis. Analisis dapat menggunakan normalisasi min-max atau standardisasi Z-score (Al-Faiz et al., 2019):

$$Min - Max\ normalisasi: X^* = \frac{X - min(X)}{Range(X)}$$

$$Standardisasi\ Z - score: X^* = \frac{X - mean(X)}{SD(X)}$$

Semua metode *clustering* memiliki tujuan untuk mengidentifikasi kelompok dari data yang tersimpan sedemikian rupa sehingga kesamaan dalam suatu kelompok sangat tinggi sedangkan kesamaan dengan data tersimpan dalam kelompok lain sangat rendah. Dengan kata lain, seperti yang ditunjukkan pada Gambar 3.1, algoritma *clustering* berusaha untuk membangun kelompok data sedemikian rupa sehingga variasi antar-cluster lebih besar dibandingkan dengan variasi dalam-cluster. Ini agak analog dengan konsep di balik analisis variansi.



Gambar 3.1 Klaster harus memiliki variasi dalam-klaster yang kecil dibandingkan dengan variasi antar–klaster (Larose & Larose, 2015)

3.2 Metode Hierarki Clustering

Algoritma *clustering* bisa bersifat hierarki atau nonhierarki. Dalam hierarki *clustering*, struktur klaster seperti pohon (dendrogram) dibuat melalui partisi rekursif (metode pembagian) atau penggabungan (*agglomerative*) dari klaster yang ada. Metode *agglomerative clustering* menginisialisasi setiap pengamatan menjadi kelompok kecilnya sendiri. Kemudian, pada langkah-langkah selanjutnya, dua cluster terdekat digabungkan menjadi cluster gabungan baru. Dengan cara ini, jumlah cluster dalam kumpulan data berkurang satu di setiap langkah. Akhirnya, semua data tersimpan digabungkan menjadi satu *cluster* besar. Metode *divisive clustering* dimulai dengan semua data tersimpan dalam satu cluster besar, dengan data yang paling berbeda dipisahkan secara rekursif, menjadi cluster terpisah, hingga setiap data tersimpan mewakili clusternya sendiri. Karena sebagian besar program komputer yang menerapkan hierarki *clustering* menggunakan metode agglomeratif, kami fokus pada metode tersebut (Larose & Larose, 2015).

Jarak antar data mudah dihitung jika pemrograman dan normalisasi yang sesuai telah dilakukan. Tapi bagaimana kita menentukan jarak antara *clustering*? Berapa jarak antar cluster yang disebut dekat?

Terdapat beberapa kriteria untuk menentukan jarak antara cluster antara A dan B:

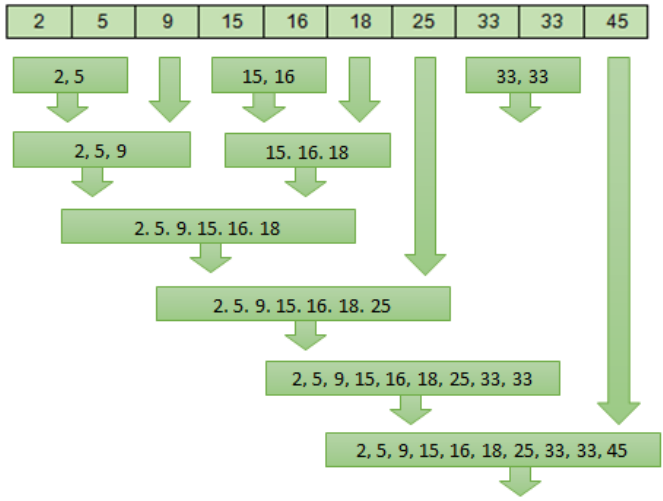
- *Single linkage*, terkadang disebut pendekatan tetangga terdekat, didasarkan pada jarak minimum antara data mana pun di cluster A dan data apa pun di cluster B. Dengan kata lain, kesamaan klaster didasarkan pada kesamaan anggota yang paling mirip dari setiap klaster. *Single linkage* cenderung membentuk kelompok yang panjang dan ramping, yang kadang-kadang menyebabkan data heterogen dikelompokkan bersama.
- *Complete linkage*, terkadang disebut pendekatan tetangga terjauh, didasarkan pada jarak maksimum antara data apa pun di cluster A dan data apa pun di cluster B. Dengan kata lain, kesamaan cluster didasarkan pada kesamaan anggota yang paling berbeda dari setiap cluster . *Complete linkage* cenderung membentuk kelompok yang lebih padat dan mirip bola.
- *Average linkage* dirancang untuk mengurangi ketergantungan kriteria *cluster-linkage* pada nilai ekstrim, seperti data yang paling mirip atau berbeda. Dalam average linkage, kriterianya adalah jarak rata-rata dari semua data dalam cluster A dari semua data dalam cluster B. Cluster yang dihasilkan cenderung memiliki variabilitas dalam cluster yang kira-kira sama.

Sebagai contoh mari kita periksa bagaimana metode *linkage* ini bekerja, dengan menggunakan kumpulan data kecil satu dimensi berikut:

2	5	9	15	16	18	25	33	33	45

3.3 Single-Linkage Clustering

Misalkan kita tertarik untuk menggunakan s*ingle linkage agglomerative* pada kumpulan data ini. Metode agglomerative dimulai dengan menugaskan setiap data ke klusternya sendiri. Kemudian, *single linkage* mencari jarak minimum antara setiap data dalam dua cluster. Gambar 3.2 mengilustrasikan bagaimana hal ini dicapai untuk kumpulan data ini. Jarak klaster minimum jelas antara klaster s*ingle linkage* di mana masing-masing berisi nilai 33, yang jaraknya harus 0 untuk setiap metrik yang valid. Dengan demikian, kedua klaster ini digabungkan menjadi klaster baru dari dua data, keduanya bernilai 33, seperti yang ditunjukkan pada Gambar 3.2. Perhatikan bahwa, setelah langkah 1, hanya tersisa sembilan (n − 1) klaster. Selanjutnya, pada langkah 2, cluster yang berisi nilai 15 dan 16 digabungkan menjadi cluster baru, karena jarak 1 mereka adalah minimum antara dua cluster yang tersisa.



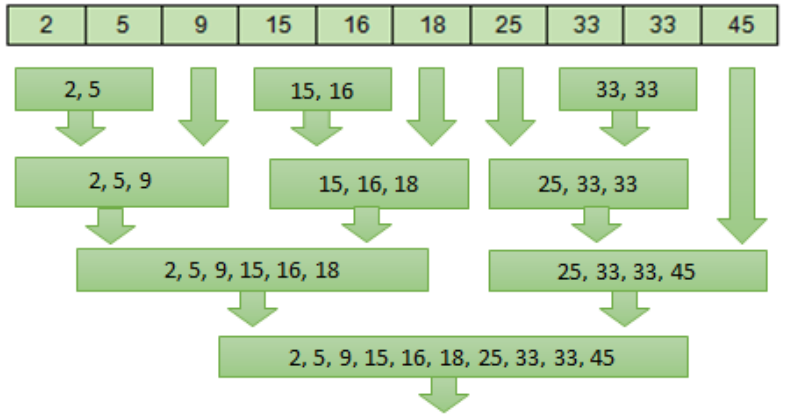
Gambar 3.2 S*ingle linkage* agglomerative clustering pada kumpulan data sampel (Larose & Larose, 2015)

Berikut adalah langkah-langkah selanjutnya:

- Langkah 3: Cluster yang berisi nilai 15 dan 16 (cluster {15,16}) digabungkan dengan cluster {18}, karena jarak antara 16 dan 18 (record terdekat di setiap cluster) adalah 2, minimum antar cluster yang tersisa .
- Langkah 4: Gugus {2} dan {5} digabungkan.
- Langkah 5: Cluster {2,5} digabungkan dengan cluster {9}, karena jarak antara 5 dan 9 (record terdekat di setiap cluster) adalah 4, jarak minimum antara cluster yang tersisa.
- Langkah 6: Cluster {2,5,9} digabungkan dengan cluster {15,16,18}, karena jarak antara 9 dan 15 adalah 6, minimum antar cluster yang tersisa.
- Langkah 7: Cluster {2,5,9,15,16,18} digabungkan dengan cluster {25}, karena jarak antara 18 dan 25 adalah 7, minimum antar cluster yang tersisa.
- Langkah 8: Cluster {2,5,9,15,16,18,25} digabungkan dengan cluster {33,33}, karena jarak antara 25 dan 33 adalah 8, minimum antar cluster yang tersisa.
- Langkah 9: Cluster {2,5,9,15,16,18,25,33,33} digabungkan dengan cluster {45}. Cluster terakhir ini sekarang berisi semua catatan dalam kumpulan data.

3.4 Complete-Linkage Clustering

Selanjutnya, mari kita periksa apakah menggunakan kriteria *complete-linkage* akan menghasilkan pengelompokan yang berbeda dari kumpulan data sampel ini (Ramadhani et al., 2018). Tautan lengkap berupaya meminimalkan jarak antar data dalam dua cluster yang terjauh satu sama lain. Gambar 3.3 mengilustrasikan *complete-linkage clustering* untuk kumpulan data ini.



Gambar 3.3 Complete-linkage agglomerative clustering pada kumpulan data sampel (Larose & Larose, 2015).

- Langkah 1: Karena setiap klaster berisi satu data saja, tidak ada perbedaan antara *single linkage* dan *complete linkage* pada langkah 1. Dua klaster yang masing-masing berisi 33 digabungkan lagi.
- Langkah 2: Sama seperti untuk *single linkage*, cluster yang berisi nilai 15 dan 16 digabungkan menjadi cluster baru. Sekali lagi, ini karena tidak ada perbedaan dalam dua kriteria untuk klaster *single* data.
- Langkah 3: Pada titik ini, hubungan lengkap mulai menyimpang dari pendahulunya. Dalam *single linkage*, klaster {15,16} pada titik ini digabungkan dengan klaster {18}. Tetapi *complete linkage* melihat tetangga terjauh, bukan tetangga terdekat.
- Tetangga terjauh untuk kedua klaster ini adalah 15 dan 18, untuk jarak 3. Ini adalah jarak yang sama yang memisahkan klaster {2} dan {5}. Kriteria *complete linkage* tidak terkait dengan ikatan, jadi kami secara acak memilih kombinasi pertama yang ditemukan, oleh karena itu menggabungkan gugus {2} dan {5} ke dalam gugus baru.

- Langkah 4: Sekarang cluster {15,16} digabungkan dengan cluster {18}.
- Langkah 5: Cluster {2,5} digabungkan dengan cluster {9}, karena jarak *complete linkage* adalah 7, terkecil di antara cluster yang tersisa.
- Langkah 6: Cluster {25} digabungkan dengan cluster {33,33}, dengan jarak *complete linkage* 8.
- Langkah 7: Cluster {2,5,9} digabungkan dengan cluster {15,16,18}, dengan *complete-linkage* distance 16.
- Langkah 8: Cluster {25,33,33} digabungkan dengan cluster {45}, dengan jarak *complete linkage* 20.
- Langkah 9: Cluster {2,5,9,15,16,18} digabungkan dengan cluster {25,33,33,45}. Semua data sekarang terkandung dalam *cluster* besar terakhir ini.

Terakhir, dengan *average linkage*, kriterianya adalah jarak rata-rata semua data di cluster A dari semua data di cluster B. Karena rata-rata satu data adalah nilai data itu sendiri, metode ini tidak berbeda dengan metode sebelumnya di tahap awal, di mana cluster single-data digabungkan. Pada langkah 3, keterkaitan rata-rata akan dihadapkan pada pilihan menggabungkan klaster {2} dan {5}, atau menggabungkan klaster {15,16} dengan klaster {18} rekaman tunggal. Jarak rata-rata antara cluster {15,16} dan cluster {18} adalah rata-rata dari $|18 - 15|$ dan $|18 - 16|$, yaitu 2,5, sedangkan jarak rata-rata antara cluster {2} dan {5} tentu saja 3. Oleh karena itu, *average linkage* akan menggabungkan cluster {15,16} dengan cluster {18} pada langkah ini , diikuti dengan menggabungkan kluster {2} dengan kluster {5}. Pembaca dapat memverifikasi bahwa kriteria *average linkage* mengarah ke struktur hierarki yang sama untuk contoh ini sebagai kriteria *complete linkage*. Secara umum, *average linkage* mengarah ke cluster yang bentuknya lebih mirip untuk melengkapi keterkaitan daripada *single linkage*.

3.5 k-Means Clustering

Algoritma k-means clustering adalah algoritma yang mudah dan efektif untuk menemukan cluster dalam data. Algoritma berlangsung sebagai berikut (Foreman, 2014):

- Langkah 1: Tanyakan kepada pengguna berapa banyak cluster k kumpulan data yang harus dipartisi.
- Langkah 2: Secara acak tetapkan k data sebagai lokasi pusat cluster awal.
- Langkah 3: Untuk setiap data, temukan pusat cluster terdekat. Jadi, dalam arti tertentu, setiap pusat klaster "memiliki" subset dari catatan, sehingga mewakili partisi dari kumpulan data. Oleh karena itu kami memiliki k cluster, $C_1, C_2, \ldots, C_k$.
- Langkah 4: Untuk setiap k cluster, temukan centroid cluster, dan perbarui lokasi setiap pusat cluster ke nilai centroid yang baru.
- Langkah 5: Ulangi langkah 3–5 hingga konvergensi atau terminasi.

Kriteria "terdekat" pada langkah 3 biasanya menggunakan jarak Euclidean, dengan nilai terkecil, meskipun kriteria lain dapat diterapkan juga. Centroid cluster pada langkah 4 ditemukan sebagai berikut. Misalkan kita memiliki n titik data (a_1, b_1, c_1), $(a_2, b_2, c_2), \ldots, (a_n, b_n, c_n)$, centroid dari titik-titik ini adalah pusat gravitasi dari titik-titik ini dan terletak di titik $\sum \frac{a_i}{n}, \sum \frac{b_i}{n}, \sum \frac{c_i}{n}$, Misalnya titik (1,1,1), (1,2,1), (1,3,1), dan (2,1,1) akan memiliki pusat massa

$$\left(\frac{1+1+1+2}{4}, \frac{1+2+3+1}{4}, \frac{1+1+1+1}{4}\right) = (1.25, 1.75, 1.00)$$

Algoritma berakhir ketika centroid tidak lagi berubah. Dengan kata lain, algoritma berhenti ketika untuk semua cluster $C_1, C_2, \ldots, C_k$, semua data yang "dimiliki" oleh masing-masing

pusat cluster tetap berada di cluster tersebut. Alternatifnya, algoritma dapat berhenti ketika beberapa kriteria konvergensi terpenuhi, seperti tidak ada penyusutan yang signifikan dalam mean squared error (MSE):

$$MSE = \frac{SSE}{N-k} = \frac{\sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)^2}{N-k}$$

dengan SSE mewakili kesalahan jumlah kuadrat, $p \in C_i$ mewakili setiap titik data dalam cluster i, m_i mewakili centroid (pusat cluster) cluster i, N adalah ukuran sampel total, dan k adalah jumlah cluster. Ingatlah bahwa algoritma *clustering* berusaha untuk membangun kelompok data sedemikian rupa sehingga variasi antar-cluster lebih besar dibandingkan dengan variasi dalam-cluster. Karena konsep ini analog dengan analisis varians, kita dapat mendefinisikan statistik pseudo-F sebagai berikut:

$$F_{k-1,N-k} = \frac{MSB}{MSE} = \frac{SSB/k-1}{SSE/N-k}$$

dengan SSE didefinisikan seperti di atas, MSB adalah kuadrat rata-rata antara, dan SSB adalah jumlah kuadrat antar cluster, didefinisikan sebagai

$$SSB = \sum_{i=1}^{k} n_i . d(m_i, M)^2$$

dengan n_i adalah jumlah data dalam cluster i, m_i adalah centroid (pusat cluster) untuk cluster i, dan M adalah rata-rata dari semua data.

MSB mewakili variasi antar-cluster dan MSE mewakili variasi dalam-cluster. Dengan demikian, cluster yang "baik" akan memiliki nilai statistik pseudo-F yang besar, mewakili situasi di mana variasi antar-cluster besar dibandingkan dengan variasi dalam-cluster. Oleh karena itu, saat algoritma k-means berjalan, dan kualitas kluster meningkat, kami berharap MSB meningkat, MSE menurun, dan F meningkat.

3.6 Contoh k-Means Clustering pada dunia kerja

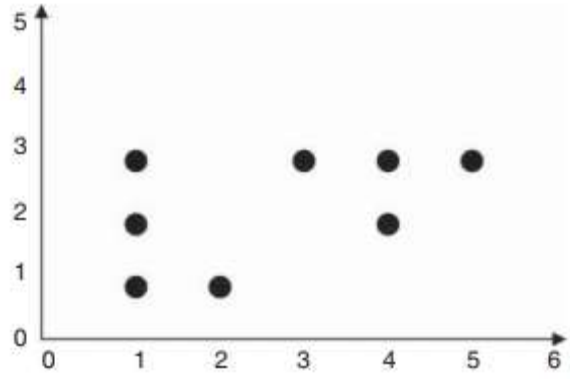
Mari kita lihat contoh bagaimana algoritma k-means bekerja. Misalkan kita memiliki delapan titik data dalam ruang dua dimensi yang ditunjukkan pada Tabel 3.1 dan diplot pada Gambar 3.4 dan tertarik untuk mengungkap k = 2 kluster.

Mari kita terapkan algoritma k-means langkah demi langkah.

- Langkah 1: Tanyakan kepada pengguna berapa banyak cluster k kumpulan data yang harus dipartisi. Misalkan cluster akan diatur pada k = 2 cluster.
- Langkah 2: Secara acak tetapkan k data sebagai lokasi pusat cluster awal. Untuk contoh ini, kami menetapkan pusat cluster menjadi $m_1 = (1,1)$ dan $m_2 = (2,1)$.
- Step 3 (first pass): Untuk setiap data, temukan pusat cluster terdekat. Tabel 3.2 berisi jarak Euclidean (dibulatkan) antara setiap titik dan setiap pusat cluster $m_1 = (1,1)$ dan $m_2 = (2,1)$, bersama dengan indikasi pusat cluster mana titik terdekat. Oleh karena itu, klaster 1 berisi titik {a,e,g}, dan klaster 2 berisi titik {b,c,d,f,h}.
- Langkah 4 (first pass): Untuk setiap k cluster, temukan centroid cluster dan perbarui lokasi setiap pusat cluster ke nilai centroid yang baru. Pusat massa Centroid untuk cluster 1 adalah [(1 + 1 + 1)/3, (3 + 2 + 1)/3] = (1,2). Pusat massa untuk cluster 2 adalah [(3 + 4 + 5 + 4 + 2)/5, (3 + 3 + 3 + 2 + 1)/5] = (3,6, 2,4). Cluster dan centroid (segitiga) pada akhir lintasan pertama ditunjukkan pada Gambar 3.5. Perhatikan bahwa m_1 telah bergerak ke atas ke pusat dari tiga titik di cluster 1, sementara m_2 telah bergerak ke atas dan ke kanan dalam jarak yang cukup jauh, ke pusat dari kelima titik tersebut di cluster 2.

Tabel 3.1 Data titik untuk contoh k-means (Larose & Larose, 2015)

a	b	c	d	e	f	g	h
(1, 3)	(3, 3)	(4, 3)	(5, 3)	(1, 2)	(4, 2)	(1, 1)	(2, 1)

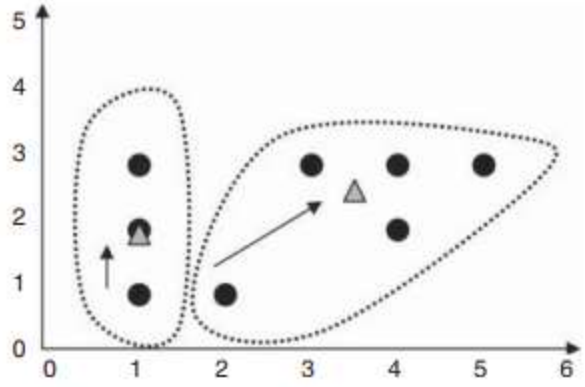


Gambar 3.4 Bagaimana k-means mempartisi data ini menjadi k = 2 cluster? (Larose & Larose, 2015)

Tabel 3.2 Menemukan pusat cluster terdekat untuk setiap data (Lulus pertama) (Larose & Larose, 2015)

Titik	**Jarak dari m_1**	**Jarak dari m_2**	**Anggota cluster**
a	2.00	2.24	C_1
b	2.83	2.24	C_2
c	3.61	2.83	C_2
d	4.47	3.61	C_2
e	1.00	1.41	C_1

f	3.16	2.24	C_2
g	0.00	1.00	C_1
h	1.00	0.00	C_2



Gambar 3.5 Cluster dan centroid Δ setelah lulus pertama melalui algoritma k-means (Larose & Larose, 2015)

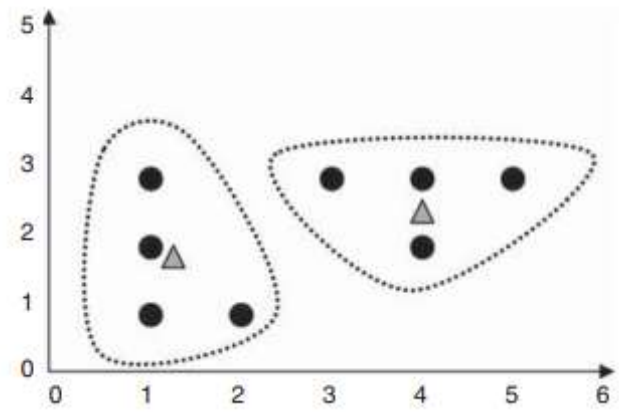
- Langkah 5: Ulangi langkah 3 dan 4 hingga konvergensi atau terminasi. Centroid telah berpindah, jadi kita kembali ke langkah 3 untuk melewati algoritme kedua.
- Step 3 (second pass): Untuk setiap record, temukan pusat cluster terdekat. Tabel 3.3 menunjukkan jarak antara setiap titik dan setiap pusat cluster yang diperbarui m_1 = (1,2) dan m_2 = (3.6, 2.4), bersama dengan keanggotaan cluster yang dihasilkan. Telah terjadi pergeseran satu record (h) dari cluster 2 ke cluster 1. Perubahan yang relatif besar dalam m2 telah meninggalkan record h sekarang lebih dekat ke m1 daripada ke m2, sehingga record h sekarang menjadi milik cluster 1. Semua record lainnya tetap berada di cluster yang sama seperti sebelumnya. Oleh karena itu, klaster 1 adalah {a,e,g,h}, dan klaster 2 adalah {b,c,d,f}

Tabel 3.3 Menemukan pusat cluster terdekat untuk setiap data (Lulus kedua) (Larose & Larose, 2015)

Titik	Jarak dari m_1	Jarak dari m_2	Anggota cluster
a	1.00	2.67	C_1
b	2.24	0.85	C_2
c	3.16	0.72	C_2
d	4.12	1.52	C_2
e	0.00	2.63	C_1
f	3.00	0.57	C_2
g	1.00	2.95	C_1
h	1.41	2.13	C_1

- Langkah 4 (Pass Kedua): Untuk masing -masing kluster K, temukan cluster centroid dan perbarui lokasi setiap pusat cluster ke nilai baru centroid. Centroid baru untuk cluster 1 adalah [(1 + 1 + 1 + 2)/4, (3 + 2 + 1 + 1)/4] = (1.25, 1.75). Centroid baru untuk cluster 2 adalah [(3 + 4 + 5 + 4)/4, (3 + 3 + 3 + 2)/4] = (4, 2.75). Cluster dan centroid pada akhir pass kedua ditunjukkan pada Gambar 19.6. Centroids M1 dan M2 keduanya bergerak sedikit.
- Langkah 5: Ulangi langkah 3 dan 4 hingga konvergensi atau penghentian. Karena centroid telah bergerak, kami sekali lagi kembali ke langkah 3 untuk ketiga (dan ternyata, final) melewati algoritma.

- Langkah 3 (Pass Ketiga): Untuk setiap catatan, temukan pusat cluster terdekat. Tabel 19.4 menunjukkan jarak antara setiap titik dan setiap pusat cluster yang baru diperbarui m_1 = (1.25, 1.75) dan m_2 = (4, 2.75), bersama dengan keanggotaan kluster yang dihasilkan. Perhatikan bahwa tidak ada catatan yang menggeser keanggotaan cluster dari pass sebelumnya.
- Langkah 4 (Pass Ketiga): Untuk masing -masing kluster K, temukan cluster centroid dan perbarui lokasi setiap pusat cluster ke nilai baru centroid. Karena tidak ada catatan yang menggeser keanggotaan cluster, centroid cluster juga tetap tidak berubah.
- Langkah 5: Ulangi langkah 3 dan 4 hingga konvergensi atau penghentian. Karena centroid tetap tidak berubah, algoritma berakhir.



Gambar 3.6 Cluster dan centroid Δ setelah lulus kedua melalui algoritma k-means (Larose & Larose, 2015)

Tabel 3.4 Menemukan pusat cluster terdekat untuk setiap data (Lulus ketiga) (Larose & Larose, 2015)

Titik	Jarak dari m_1	Jarak dari m_2	Anggota cluster
a	1.27	3.01	C_1
b	2.15	1.03	C_2
c	3.02	0.25	C_2
d	3.95	1.03	C_2
e	0.35	3.09	C_1
f	2.76	0.75	C_2
g	0.79	3.47	C_1
h	1.06	2.66	C_1

3.7 Perilaku MSB, MSE, dan Pseudo-F sebagai pemroses Algoritma k-Means

Mari kita amati perilaku statistik ini setelah langkah 4 dari setiap kelulusan.

Lulus pertama:

- $SSB = \sum_{i=1}^{k} n_i \cdot d(m_i, M)^2 = 3 \cdot d\big((1,2), (2.625, 2.25)\big)^2 + 5 \cdot d\big((3.6,2.4), (2.625, 2.25)\big)^2 = 12.975$
- $MSB = \frac{SSB}{k-1} = \frac{12.975}{2-1} = 12.975$
- $SSE = \sum_{i=1}^{k} \sum_{p \epsilon C_i} d(p, m_i)^2 = 2^2 + 2.24^2 + 2.83^2 + 3.61^2 + 1^2 + 2.24^2 + 0^2 + 0^2 = 36$

- $MSE = \frac{SSE}{N-k} = \frac{36}{6} = 6$
- $F = \frac{MSB}{MSE} = \frac{12.975}{6} = 2.1625$

Secara umum, kami mengharapkan MSB meningkat, MSE menurun, dan F meningkat, dan begitulah yang terjadi. Perhitungan dibiarkan sebagai latihan.

Lulus kedua: MSB = 17.125, MSE = 1.313333, F = 13.03934.

Lulus ketiga: MSB = 17.125, MSE = 1.041667, F = 16.44.

Statistik ini menunjukkan bahwa telah dicapai variasi antar-cluster maksimum (sebagaimana diukur dengan MSB), dibandingkan dengan variasi dalam-cluster (sebagaimana diukur dengan MSE).

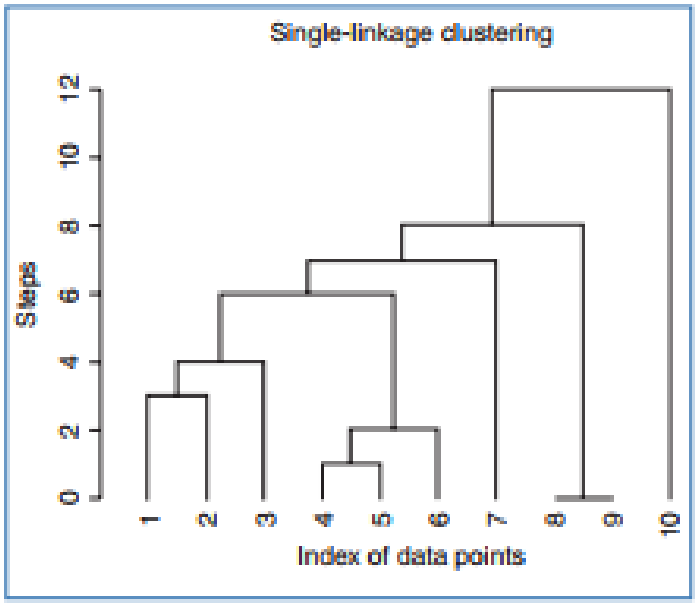
Perhatikan bahwa algoritma k-means tidak dapat menjamin penemuan statistik pseudo-F maksimum global, alih-alih sering ditetapkan pada maksimum lokal. Untuk meningkatkan kemungkinan mencapai minimum global, analisis dapat mempertimbangkan untuk menggunakan berbagai pusat cluster awal. Moore menyarankan (i) menempatkan pusat cluster pertama secara acak titik data, dan (ii) menempatkan pusat klaster berikutnya pada titik-titik yang sejauh mungkin dari pusat sebelumnya.

Satu masalah potensial untuk menerapkan algoritma k-means adalah: Siapa yang memutuskan berapa banyak cluster yang akan dicari? Artinya, siapa yang memutuskan k? Kecuali jika analis memiliki pengetahuan sebelumnya tentang jumlah klaster yang mendasarinya; oleh karena itu, "loop luar" harus ditambahkan ke algoritme, yang menggilir berbagai nilai k yang menjanjikan. Solusi pengelompokan untuk setiap nilai k karenanya dapat dibandingkan, dengan nilai k menghasilkan statistik F terbesar yang dipilih. Sebagai alternatif, beberapa algoritme pengelompokan, seperti algoritme pengelompokan BIRCH, dapat memilih jumlah klaster yang optimal.

Bagaimana jika beberapa atribut lebih relevan daripada yang lain untuk perumusan masalah? Karena keanggotaan klaster ditentukan oleh jarak, kita dapat menerapkan metode peregangan sumbu yang sama untuk menghitung relevansi atribut yang telah kita bahas di Bab 10. Di Bab 20, kita memeriksa metode pengelompokan umum lainnya, jaringan Kohonen, yang terkait dengan jaringan syaraf tiruan di struktur.

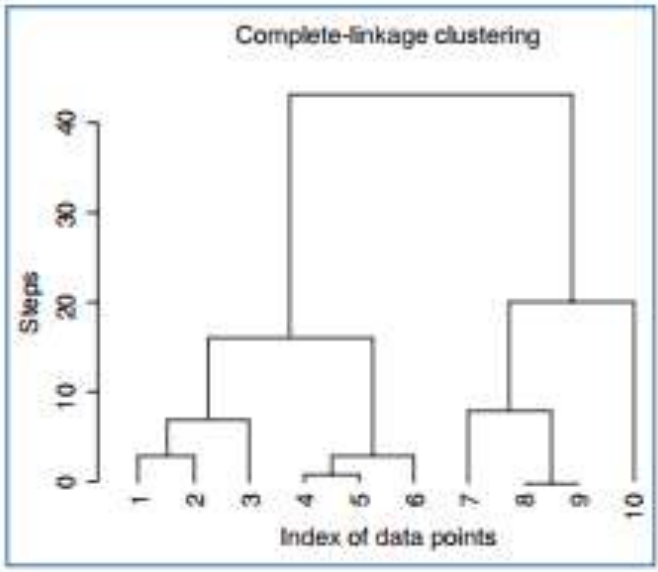
3.8 Zona R

```
# Instal paket yang diperlukan dan buat datanya

library(cluster)

data <- c(2, 5, 9, 15, 16, 18, 25, 33, 33, 45)

# Single-Linkage Clustering
agn <- agnes(data,
        diss = FALSE,
        stand = FALSE,
        method = "single")
# Make and plot the dendrogram
dend_agn <- as.dendrogram(agn)
plot(dend_agn,
        xlab = "Index of Data Points",
        ylab = "Steps",
        main = "Single-Linkage Clustering")
```



Gambar 3.7 Single Linkage Clustering (Larose & Larose, 2015)

```
# Complete-Linkage Clustering
agn_complete <- agnes(data,
        diss = FALSE,
        stand = FALSE,
        method = "complete")

# Make and plot the dendrogram
dend_agn_complete <-
        as.dendrogram(agn_complete)
plot(dend_agn_complete,
        xlab = "Index of Data Points",
        ylab = "Steps",
        main = "Complete-Linkage Clustering")
```



Gambar 3.8 Complete Linkage Clustering (Larose & Larose, 2015)

K-Means clustering

```
# Create the data matrix
# from Table 10.1
m <- matrix(c(1,3,3,3,4,3,5,3,1,2,4,2,1,1,2,1),
            byrow=TRUE,
            ncol = 2)
km <- kmeans(m,centers = 2)
km
```

Hasilnya:

```
>km
K-means clustering with 2 clusters of sizes 4, 4
Cluster means:
    [,1]  [,2]
1  1.25  1.75
```

2 4.00 2.75

clustering vector:

a b c d e f g h

1 2 2 2 1 2 1 1

within cluster sum of squares by cluster:

[1] 3.50 2.75

(between_ss / total_ss = 73.3 %)

Available components:

[1] “cluster” “centers” “totss”

[2] “withinss” “tot.withinss” “betweenss”

[3] “size”

Bab 4

SQL Basis Data

4.1 Pengantar

Di era teknologi informasi saat ini banyak bidang yang tidak dapat dipisahkan dari basis data. Mulai dari bisnis retail, rumah sakit, perguruan tinggi, perusahaan swasta, pemerintahan sampai kepada militer, semua membutuhkan basis data untuk menyimpan data-data penting mereka. Penggunaan basis data yang semakin populer turut mendongkrak kepopuleran SQL karena keduanya memang tidak bisa dipisahkan. Tidak ada SQL tanpa adanya basis data, dan begitu pula sebaliknya.

SQL adalah bahasa standar untuk mengakses dan memanipulasi basis data yang merupakan singkatan dari Structured Query Language. SQL menjadi bahasa standar dan sudah disahkan oleh American National Standards Institute (ANSI) dan International Organization for Standardization (ISO).

SQL memiliki kemampuan dan mempunyai fungsi diantaranya :

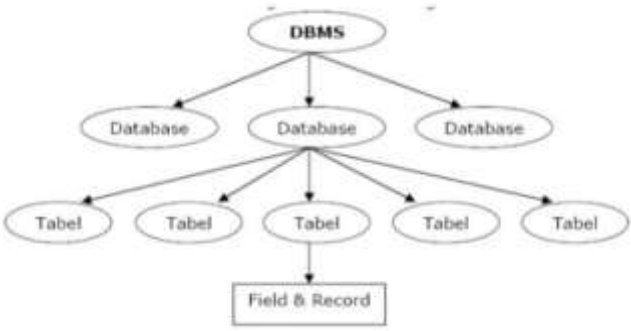
1. Memiliki kemampuan dan mempunyai fungsi untuk membuat basis data baru
2. Memiliki kemampuan dan mempunyai fungsi untuk membuat tabel baru dalam basis data

3. Memiliki kemampuan dan mempunyai fungsi untuk menyisipkan data ke dalam basis data
4. Memiliki kemampuan dan mempunyai fungsi untuk mengambil data dari basis data
5. Memiliki kemampuan dan mempunyai fungsi untuk memperbarui catatan dalam basis data
6. Memiliki kemampuan dan mempunyai fungsi untuk menghapus catatan dari basis data
7. Memiliki kemampuan dan mempunyai fungsi untuk mengeksekusi query terhadap basis data
8. Memiliki kemampuan dan mempunyai fungsi untuk membuat prosedur tersimpan dalam basis data
9. Memiliki kemampuan dan mempunyai fungsi untuk membuat tampilan dalam basis data

Meskipun SQL adalah standar ANSI/ISO, ada beberapa versi bahasa SQL yang berbeda. Namun, agar sesuai dengan standar ANSI, semuanya mendukung setidaknya perintah utama (seperti SELECT, UPDATE, DELETE, INSERT, WHERE) dengan cara yang sama.

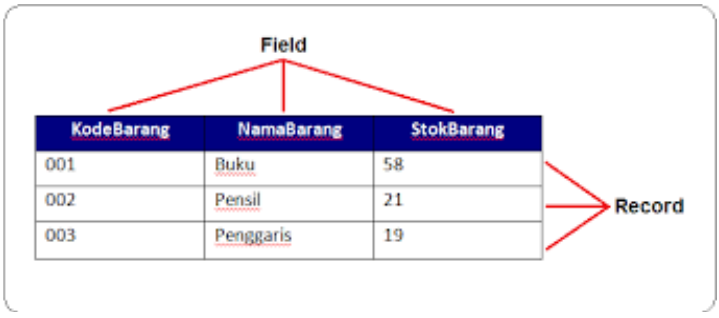
Untuk membuat website yang menampilkan data dari basis data, Kita memerlukan perangkat lunak basis data RDBMS seperti Microsoft Access, Oracle, PostgreeSQL, Microsoft SQL atau MySQL, menggunakan bahasa skrip PHP atau ASP dan menggunakan HTML dan CSS untuk tata letak halaman website.

Relation Basis data Management System (RDBMS) adalah sistem basis data modern seperti Microsoft Access, Oracle, PostgreeSQL, Microsoft SQL atau MySQL. Data dalam RDBMS disimpan dalam objek basis data yang disebut tabel. Tabel adalah kumpulan field yang saling terkait dan terdiri dari kolom dan baris.



Gambar 4.1 Struktur Basis data

Field adalah pecahan dari tabel yang menjadi entitas yang lebih. Contoh field dalam tabel barang terdiri dari KodeBarang, NamaBarang, StokBarang. Field adalah kolom dalam tabel yang dirancang untuk menyimpan informasi tentang setiap catatan dalam tabel. Record adalah setiap entri individu data yang ada dalam tabel disetiap barisnya. Record adalah entitas baris dalam tabel.



Gambar 4.2 Tabel, Field dan Record

4.2 Sintak SQL

Basis data paling sering berisi satu atau lebih tabel. Setiap tabel identifikasi dengan nama (misalnya "Pelanggan" atau "Pesanan"). Tabel berisi record baris) dengan data. Dibawah ini adalah contoh record tabel pelanggan

Pelanggan

PelangganID	NamaPelanggan	AlamatPelanggan	Kota	KodePos	Negara
1	John Friadi	Obere Str. 57	Berlin	12209	Germany
2	Lauren Eka Wijaya	Avda. de la Constitución 2222	Mexico	05021	Mexico
3	Faris Raihan	Mataderos 2312	Mexico	05023	Mexico
4	Faiq Raihan	120 Hanover Sq.	London	12210	UK

Gambar 4.3 Record Tabel Pelanggan

Tabel di atas berisi empat record (satu untuk setiap pelanggan) dan enam kolom (PelagganID, NamaPelanggan, Alamat Pelanggan, Alamat, Kota, Kode Pos, dan Negara). Sebagian besar tindakan yang perlu Anda lakukan pada basis data dilakukan dengan pernyataan SQL. Pernyataan SQL berikut memilih semua catatan dalam tabel "Pelanggan":

Contoh:

SELECT * FROM Customers;

Berikut adalah ketentuan-ketentuan memberi perintah pada SQL:

1. Setiap perintah harus diakhiri dengan tanda titik koma, kecuali untuk perintah tertentu, misalnya : quit
2. Perintah-perintah dalam lingkungan QL tidak menerapkan aturan case sensitive, case insensitive yaitu perintah bisa dituliskan dalam huruf besar atau pun huruf kecil.
3. Beberapa perintah SQL membutuhkan tanda “;” (titik koma) disetiap di akhir pernyataan SQL.
4. Tanda “;” (titik koma) adalah cara untuk memisahkan setiap pernyataan SQL dalam sistem.

Berikut ini adalah perintah-perintah SQL yang sering digunakan di dalam pengelolaan basis data diantaranya :

1. Perintah SELECT - memilih data dari basis data
2. Perintah UPDATE - memperbarui data dalam basis data

3. Perintah DELETE - menghapus data dari basis data
4. Perintah INSERT INTO - memasukkan data baru ke dalam basis data
5. Perintah CREATE BASIS DATA - membuat basis data baru
6. Perintah ALTER BASIS DATA - memodifikasi basis basis data
7. Perintah CREATE TABLE - membuat tabel baru
8. Perintah ALTER TABLE - memodifikasi tabel
9. Perintah DROP TABLE - menghapus tabel

Sintak SQL SELECT

Pernyataan SELECT berfungsi untuk memilih record/data dari basis data.

Sintaknya :

SELECT column1, column2, ...
FROM table_name;

Column1, column2, ... adalah nama field dari tabel yang akan dipilih datanya.

Contoh :

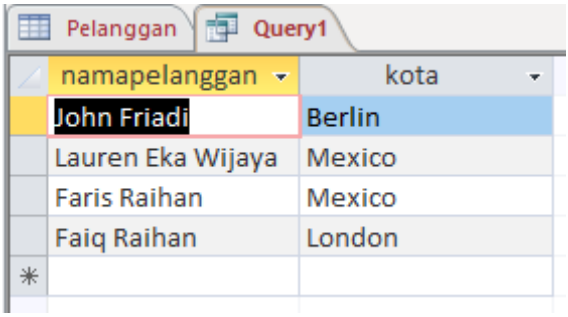
Memilih kolom "Nama Pelanggan" dan "Kota" dari tabel "Pelanggan":

Sintaknya :

SELECT namapelanggan, kota FROM pelanggan;

Maka yang tampil akan seperti gambar dibawah ini :

Hasilnya :



Gambar 4.4 Menampilkan Nama Pelanggan dan Kota dengan SQL

Sintak SQL WHERE

Sintak WHERE berfungsi untuk menyaring record/data yang memenuhi kondisi tertentu.

Sintaknya :

SELECT column1, column2, ...
FROM table_name
WHERE condition;

Keterangan :

Sintak WHERE juga dapat digunakan untuk perintah UPDATE, DELETE, dan lain-lain.

Contoh :

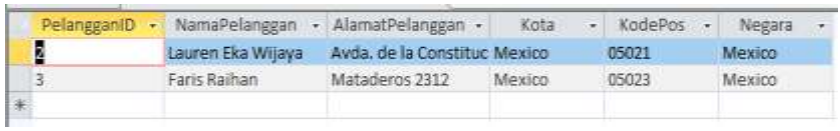
Perintah memilih seluruh data pelanggan dari negara "Meksiko" saja dari tabel "Pelanggan":

Sintaknya :

SELECT * FROM Pelanggan

WHERE Negara='Mexico';

Hasilnya :



PelangganID	NamaPelanggan	AlamatPelanggan	Kota	KodePos	Negara
2	Lauren Eka Wijaya	Avda. de la Constituc	Mexico	05021	Mexico
3	Faris Raihan	Mataderos 2312	Mexico	05023	Mexico

Pada sintak SQL WHERE ada operator yang sering digunakan selain =. Berikut ini adalah operator-operator yang dapat digunakan dalam perintah SQL WHERE

Operator	Keterangan
=	Sama dengan
>	Lebih besar
<	Lebih kecil
>=	Lebih besar sama dengan
<=	Lebih kecil sama dengan
<>	Tidak sama Dalam beberapa versi SQL, ditulis !=
BETWEEN	Diantara rentang tertentu
LIKE	Mencari pola
IN	Untuk menentukan beberapa kemungkinan nilai untuk kolom

4.3 Sintak SQL AND, OR dan NOT

Klausa WHERE dapat digabungkan dengan perintah AND, OR, dan NOT.

1. Menampilkan data lebih dari satu kondisi dapat menggunakan Operator AND dan OR.

2. Menampilkan data jika semua kondisi adalah TRUE dapat menggunakan Operator AND.
3. Menampilkan data jika salah satu kondisi adalah TRUE dapat menggunakan Operator OR.
4. Operator NOT menampilkan data jika kondisi NOT TRUE.

Sintak AND :

SELECT column1, column2, ...

FROM table_name
WHERE condition1 AND condition2 AND condition3 ...

Contoh AND :

Memilih semua data dari tabel "Pelanggan" dengan negara adalah "Jerman" DAN kota adalah "Berlin".

Sintaknya :

SELECT * FROM Pelanggan
WHERE Negara='Germany' AND Kota='Berlin'

Sintak OR :

SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;

Contoh OR :

Memilih semua data dari tabel "Pelanggan" dengan kota "Berlin" ATAU "Munchen"

Sintaknya :

SELECT * FROM Pelanggan
WHERE Kota='Berlin' OR Kota='Munchen'

Sintak NOT :

SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;

Contoh NOT :

Memilih semua data dari tabel "Pelanggan" yang BUKAN negara "Germany".

Sintaknya :

SELECT * FROM Pelanggan
WHERE NOT Negara='Germany';

4.4 Sintak SQL ORDER BY

Perintah ORDER BY bertujuan untuk mengurutkan data baik secara urutan menaik atau menurun. Default perintah ORDER BY adalah mengurutkan data secara urutan menaik. Untuk mengurutkan data secara menurun, maka gunakan perintah DESC.

Sintak ORDER BY :

SELECT column1, column2, ...

FROM table_name
ORDER BY column1, column2, ... ASC|DESC;

Contoh :

Memilih semua pelanggan dari tabel "pelanggan", diurutkan sesuai kolom "Negara".

Sintaknya :

SELECT * FROM Pelanggan
ORDER BY Negara

4.5 Sintak SQL INSERT INTO

Pernyataan INSERT INTO digunakan untuk menyisipkan record baru ke dalam tabel.

Sintaks INSERT INTO

Dimungkinkan untuk menulis pernyataan INSERT INTO dengan dua cara:

1. Tentukan nama kolom-kolom dan nilai-nilai yang akan ditambahkan:

   INSERT INTO table_name (column1, column2, column3, ...)

   VALUES (value1, value2, value3, ...);

2. Jika Anda menambahkan nilai-nilai untuk semua kolom tabel, Anda tidak perlu menentukan nama-nama kolom. Namun, pastikan bahwa urutan nilai-nilai dalam urutan yang sama.

   Berikut sintaks:

   INSERT INTO table_name
   VALUES (value1, value2, value3, ...);

   Contoh :

   Menyisipkan record/data di file "Pelanggan"

   Sintaknya:

   INSERT INTO Pelanggan (NamaPelanggan, AlamatPelanggan, Kota, KodePos, Negara)

   VALUES ('Lauren', 'Volvo 21', 'Stavanger', '4004', 'Norwegia');

4.6 Sintak SQL SELECT

Berfungsi untuk menampilkan data dari basis data

Sintak SQL SELECT :

SELECT column1, column2, ...

FROM table_name

WHERE conditions

ORDER BY expression ASC | DESC;

Contoh :

SELECT * FROM Pelanggan
SELECT NamaPelanggan, Kota FROM Pelanggan;

Bab 5

R For Data Scientist

5.1 Pengenalan R Programmin

Globalisasi mempercepat perubahan teknologi informasi menuju hal yang positif dan berkembang juga perangkat lunak pengolahan dan analisa data, baik itu bersifat komersial maupun *open sources.* Berikut beberapa perangkat lunak yang sering digunakan dalam pengolahan dan analisa data yaitu SPSS, R, SAS, Excel dan sebagainya. Perangkat lunak "R" merupakan aplikasi *open sources* berbasis bahasa pemrograman yang digunakan untuk pengolahan data dan analisis statistika yang menggunakan *Graphic User Interface* (GUI) (Yeli, 2017) yang sangat bermanfaat bagi industri dan penelitian (Widodo, 2013).

Awal mulanya, perangkat lunak R hadir setelah pengembangan aplikasi S dan S plus. Tahun 1995, dua akademisi yakni Robert Gentleman dan Ross Ihaka dari Departemen Statistika, Universitas Auckland telah mengembangkan perangkat lunak R. Kini aplikasi R dikembangkan oleh Tim Pengembangan Peduli R (R *Development Core Team*). Kebutuhan penggunaan di industri dan penelitian untuk analisis data maka perangkat lunak R dapat dijalankan berbagai multiplatform *operating system* seperti LINUX/UNIX, Windows, dan Macintosh (Verzani, 2018). Selain itu, R memiliki GUI yang bagus, aplikasi yang *powerfull* untuk pengolahan dan analisis data, bahasa pemrograman yang

mudah dipahami dan didukung *package* atau *library* di dalam perangkat lunak R yang mantab untuk Data Science (McCreight, 2012).

Perangkat lunak R digunakan untuk *extraction, organizing, visualizing, modelling,* dan *performing* (Chamber, 2008). Sumber informasi utama mengenai seputar R secara menyeluruh dapat mengakses *Comprehensive R Archive Network* (CRAN). Aplikasi R menjadi alat pengolahan data berbasis statistika yang digemari oleh akademisi dan peneliti. Berikut alasan kenapa aplikasi R digemar, yaitu:

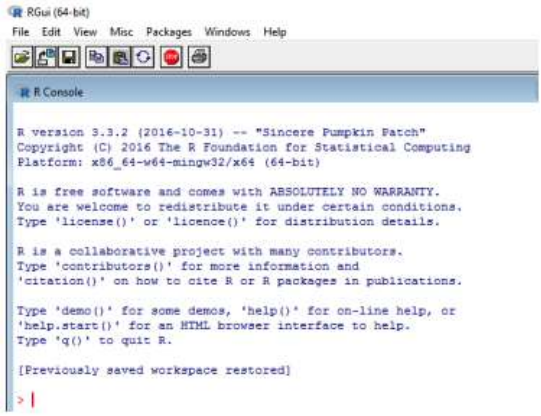
1) ***Open Sources and Free***, aplikasi R sering dapatkan bantuan dari komunitas dan sumbangan pemikiran code yang dapat dimodifikasi, diperiksa, ditambahakan dan dibagikan oleh atau dari pengguna. Seluruh dunia banyak pengguna kontribusi untuk pengembangan R.
2) ***Popularity***, Menurut lembaga riset Tiobe Index bahwa aplikasi R Programming menempati urutan ke-12 per Oktober 2021 jauh lebih meningkat daripada tahun sebelumnya (Tiobe, 2022). Aplikasi R masih diminati dan digunakan oleh perusahaan besar seperti Facebook, Google, Bing, Accenture dan Wipro untuk kebutuhan riset dan analisis data perusahaan (data-flair, 2018).
3) ***Powerfull***, R memiliki banyak *library* atau *package* yang sangat lengkap dan menjadi kekuatan aplikasi R. Misalkan penggunaan *package "ggplot"* digunakan para data scientist untuk menampilkan visualisasi dan interpretasi data (geospasialis, 2020).
4) ***Reproducible***, skrip (*coding*) R yang tersimpan mudah digunakan kembali pada saat melakukan proses analisis yang sama dan data yang berbeda (Edanz, 2018)

Aplikasi R dapat diunduh pada website https://cran.r-project.org/ sesuaikan dengan *operating system* yang digunakan (pada pembahasan selanjutnya, penulis menerangkan *code*-R yang dijalankan di platform MacOS). Apabila sistem operasi yang diinstall laptop itu adalah Windows, pilihlah *"Download R for Windows"* seperti yang ditunjukkan pada Gambar 5.1.

Gambar 5.1 Halaman Utama CRAN (Sumber : www.cran.r-project.org)

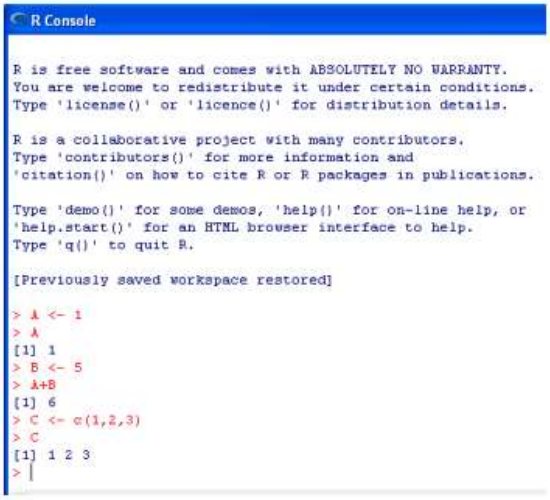
Setelah instalansi maka akan tampil dihalaman desktop komputer berupa editor dasar R yang sering dikenal sebagai dengan nama *RGui* pada *operating system* Windows yang ditunjukkan pada Gambar 5.2 , sedangkan pada MacOS sering dikenal dengan nama *R.app*. R *Console* ada dalam di RGui yang digunakan untuk tempat penulisan dan menjalankan perintah baris code. Diberikan beberapa contoh program sederhana pada R*Console* sebagai berikut:

```
A <- 1
B <- 5
A+B
C <- c(1,2,3)
```



Gambar 5.2 R Console (Sumber : Prana & Adhitya, 2017)

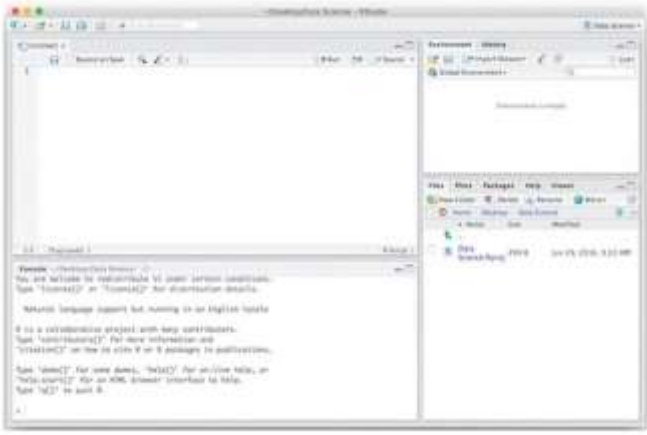
Pada Gambar 5.2 pada RConsole ada simbol ">" sering disebut sebagai tanda *prompt* yang berfungsi untuk penulisan baris *code* bahasa R. Contoh program diatas telah dijalankan (*running*) maka hasilnya akan ditunjukkan pada Gambar 5.3.



Gambar 5.3 Hasil *Running Program* (Sumber : Prana & Adhitya, 2017)

Hasil baris *code* yang telah dijalankan menampilkan simbol "[ ]" yang merupakan bentuk *output* eksekusi perintah *code* bahasa R. Aplikasi RStudio menjadi alternatif penggunaan

aplikasi R yang memiliki tampilan *Integrated Development Environment* (IDE) lebih menarik dibandingkan dengan RGui. Halaman utama RStudio yang ditunjukkan pada Gambar 5.4 terdapat *frame-frame* yang memiliki fungsinya masing-masing. Bagian *frame* kiri menerangkan sebagai area *console* yang digunakan untuk *write*, *read*, *evaluate* dan *print* . Bagian *frame* kanan atas menerangkan sebagai *environment* yang berfungsi untuk menetapkan nilai atribut atau variabel yang telah dieksekusi dari baris *code* bahasa R. Bagian *frame* kanan bawah menerangkan tempat keberadaan lembar kerja (*project*) pengguna dan lokasi file yang telah dibuat. (Thomas, 2022).



Gambar 5.4 Halaman RStudio (Thomas, 2022).

Setelah menginstal RStudio di komputer, sebagai langkah perdana penggunaan aplikasi R dan mengenal bahasa pemrograman R, selanjutnya dapat diketikkan dan jalankan perintah R pada *frame console* sebagai berikut:

```
print("Hello World")
```

Bahasa R yang telah diketikkan dengan fungsi *print* akan mencetak Hello World yang ditunjukkan pada Gambar 5.5.

Gambar 5.5 Hasil Eksekusi Fungsi Print

Perlu diperhatikan dalam pembuatan perintah R, karena bahasa R merupakan bahasa pemrograman *case-sensitive* yang artinya beda penulisan dengan huruf besar atau huruf kecil sangat mempengaruhi makna arti. Selanjutnya dapat diketikkan perintah bahasa R sebagai berikut:

```
A <- 1
print(a) #mencetak nilai objek 'a'
```

Hasil eksekusi perintah bahasa R di atas akan menampilkan "*Error in evaluating the argument x …"* hal ini disebabkan fungsi *print* tidak menampilkan isi nilai dari obyek atau variabel 'a' karena bahasa pemrograman R sangat sensitive dalam penulisannya. Seharusnya isi object dari fungsi *print* adalah variabel A.

Gambar 5.6 Error Case-Sensitive

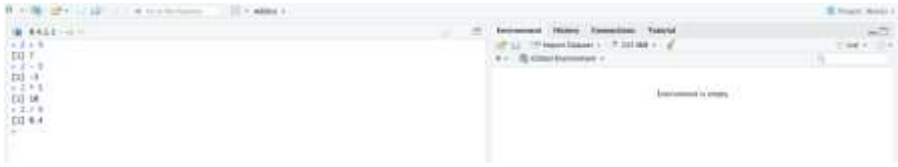
5.2 Dasar Pemrograman R

Kalkulator Sederhana

Alangkah baiknya mengupas dasar pemrograman R sebelum memasuki tingkat kesulitan pada subbab selanjutnya. Untuk dapat memahami lebih lanjut, bahasa R merupakan aplikasi berbasis statistika namun bahasa R juga dapat Kalkulator sederhana. Pada bagian *frame console* dapat diketikkan perintah bahasa R dengan operator + (penjumlahan), - (pengurangan), * (perkalian), / (pembagian) sebagai berikut:

```
2 + 5 #operator penjumlahan
2 - 5 #operator pengurangan
2 * 5 #operator perkalian
2 / 5 #operator pembagian
```

Selanjutnya jalankan (*running*) program yang telah diketikkan untuk mengetahui hasil dari kalkulator sederhana.



Gambar 5.7 Hasil Eksekusi Kalkulator Sederhana

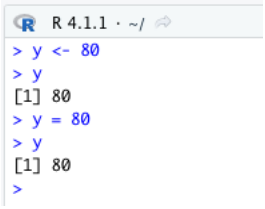
Pada Gambar 5.7 muncul hasil eksekusi kalkulator sederhana yang sudah sesuai dengan perhitungan manual bahwa hasil 2+5 = 5, 2-5=-3, 2*5=10, 2/5 = 0,4. Kemudian terdapat tanda [1] 7 dan seterusnya, tanda "[1]" menunjukkan dari

elemen pertama perintah R. Sedangkan tanda '>' merupakan tanda *prompt* yang artinya siap menerima perintah bahasa R yang baru (bookdown , 2022).

Tanda *Assignment* (← , =)

Bahasa R memiliki perbedaan dengan bahasa pemrograman yang lainnya, salah satu yang membedakan yaitu operator *assignment*. Sebagian besar, menggunakan operator "=" (sama dengan). Seringnya pengguna R menggunakan operator " ← " (panah kiri) dengan format penulisan *obj* ← *expr* artinya nilai *expr* dimasukkan ke *obj*. Pada bagian *frame console* dapat diketikan dengan perintah bahasa R, sebagai berikut:

```
Y <- 80 #menggunakan assignment <-
Y = 80 #menggunakan assignment =
```



Gambar 5.8 Tanda operator Assignment

Hasil *running* eksekusi pada Gambar 5.8 telah menunjukkan bahwa tanda operator *assignment* " <- " atau " = " akan menunjukkan isi nilai *y* tidak mengalami perubahan. Secara umum, penulisan bahasa R lebih sering menggunakan tanda operator " <- " namun kembali *programmer* (pengguna) lebih nyaman tanda "<-" atau "=". Berikut beberapa tanda *assignment* yang dapat digunakan, yaitu:

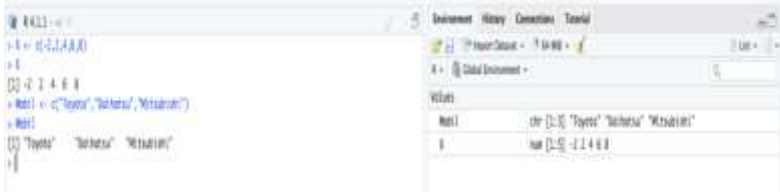
Tabel 5.1 Beberapa Tanda Operator *Assignment*

Operator	Penjelasan
<-	nilai objek dimasukkan dari sebelah kanan
->	nilai objek dimasukkan dari sebelah kiri
<<-	nilai objek global dimasukkan dari sebelah kanan
->>	nilai objek global dimasukkan dari sebelah kiri

Vektor

Tipe data dasar di R yaitu vektor, sebagai kumpulan nilai terurut yang tersusun dari nilai tunggal untuk angka, logika dan string. Vektor merupakan struktur data satu dimensi dan elemen didalamnya memiliki tipe data yang sama. Cara mudah membuat vektor menggunakan fungsi c ( ), seperti contoh berikut: (Omar, 2017)

```
X <- c(-2,2,4,6,8) # vector numeric
X
Mobil <- c("Toyota","Daihatsu","Mitsubishi")
Mobil
```



Gambar 5.9 Hasil Eksekusi Vektor

Pada Gambar 5.9 menunjukkan hasil eksekusi vektor dari dua tipe yaitu vektor *numeric* dan vektor *character.* Nilai objek dengan tipe data *character* (*string*) dalam penulisan di bahasa

pemrograman R harus ditandai dengan tanda petik " ", hal ini pun juga dilakukan dalam bahasa pemrograman yang lainnya apabila nilai objek atau variabel dengan tipe data *string* atau *character*.

Saat eksekusi bahasa pemrograman R, sistem IDE akan membaca dan mencantumkan variabel ke dalam *environment* IDE pada bagian *frame* kanan atas yang berisi variabel X dan Mobil. Variabel X akan terbaca dan tersimpan berupa *num [1:5]* : -2 2 4 6 8, tanda *num* [1:5] menunjukkan variabel X memiliki tipe data *numeric* dengan nilainya sebanyak 5 angka yaitu -2 2 4 6 8. Variabel Mobil akan terbaca dan tersimpan berupa *chr* [1:3] : "Toyota" "Daihatsu" "Mitsubishi", tanda *chr* [1:3] menunjukkan variabel Mobil memiliki tipe data *character* atau *string* dengan nilainya sebanyak 3 *character* yaitu Toyota, Daihatsu, Mitsubishi.

Factor

Saat analisis data, aplikasi pemrograman R menyediakan dengan baik untuk mengenali nilai kategori dengan menggunakan fungsi *factor* ( ) (Omar, 2017). Unsur pada *factor* ( ) harus nilai kategori atau kelas dan bukan nilai *numeric* atau angka, misal kategori "*small", "medium",* dan "*large"* (Thomas, 2017).

```
kategori                                              <-
factor(c("small","small","medium",

"medium","large","large","small")
kategori
```

Gambar 5.10 Hasil Eksekusi Factor

Nilai-nilai yang terinputkan pada objek atau variabel "kategori" dengan fungsi *factor* ( ) akan menampilkan fungsi *levels* ( ) yang menunjukkan semua level dan parameter faktor yang dapat digunakan secara eksplisit menentukan urutannya (Omar, 2017) seperti yang ditunjukkan pada Gambar 5.10 bahwa nilai dari variabel "kategori" memiliki 3 level (*"large", "medium", "small"*) dan berurutan dari terbesar sampai terkecil atau pun sebaliknya.

Metrics

Objek atau variabel pada bahasa pemrograman R memiliki 2 dimensi yaitu *row* (baris) dan *column* (kolom) serta mempunyai tipe data yang sama sering dikenal dengan istilah *metrics*. Pada saat membuat elemen *metrics* minimal terdapat 1 elemen saja dengan tipe data *character* maka *metrics* tersebut memiliki tipe data sebagai *character*. *Metrics* di bahasa R mengunakan vektor yang akan dikonversi ke dalam dimensi yang bisa dibayangkan dengan ukuran *m* baris dan 1 kolom. Misalkan kita memiliki vektor *numeric* x dengan banyak elemen sebanyak 10 buah sebagai berikut:

```
x <- c(0,1,3,2,6,4,8,7,9,5) #vektor numerik x
x
```

Hasil eksekusi vektor *numeric* x ditunjukan pada Gambar 5.11

```
R 4.1.1 · ~/
> x <- c(0,1,3,2,6,4,8,7,9,5)
> x
 [1] 0 1 3 2 6 4 8 7 9 5
```

Gambar 5.11 Hasil Eksekusi Vektor *Numeric*

Vektor *numeric* x memiliki 10 elemen, maka dimensi *metrics* yang bisa dibuat menjadi 2 angka yang hasil perkalian 5 x 2 = 10. Dapat menggunakan fungsi *matrix* ( ) dengan penentuan ukuran matriks dapat gunakan *nrow* atau *ncol*. Bisa gunakan salah satunya atau keduanya. Misalkan membuat *metrics* dengan row = 5.

```
mc <- matrix(data= x, nrow = 5)
mc
```

```
> mc <- matrix(data = x,nrow = 5)
> mc
     [,1] [,2]
[1,]    0    4
[2,]    1    8
[3,]    3    7
[4,]    2    9
[5,]    6    5
```

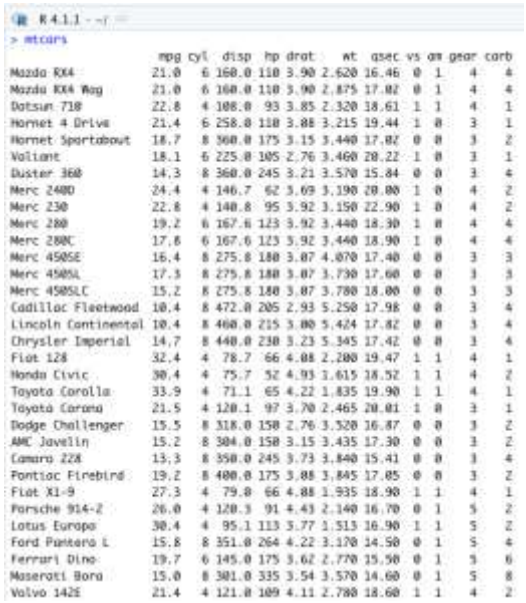
Gambar 5.12 Hasil Eksekusi Matrix

Hasil sebelumnya bahwa vektor *numeric* x adalah 0,1, 3, 2, 6, 4, 8, 7, 9, 5. 5 angka pertama dan terakhir pada vektor *numeric* x akan menjadi kolom ke-1 dan ke-2. Pada Gambar 1.12 terdapat tanda [, 1] dan [, 2] memiliki arti sebagai kolom ke-1 dan ke-2 kemudian ada tanda [1,] dan seterusnya itu memiliki arti sebagai baris ke-1 sampai ke-5 karena pada saat penulisan bahasa R sudah dideklarasikan nrow = 5 artinya *metrics* memiliki 5 baris.

Dataframe

Berkaitan matrix teringat dengan aplikasi Microsoft Excel yang memiliki tabel terdiri dari kolom dan baris. Pada bahasa pemrograman R ada istilah yang disebut dengan dataframe yang berbentuk tabel seperti aplikasi Microsoft Excel.

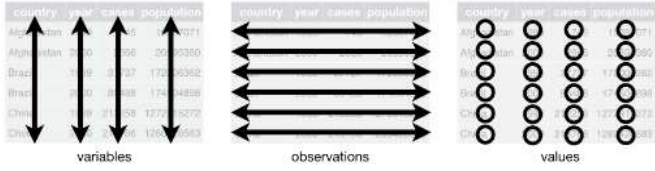
Perbedaan yang mendasar antara matrix dengan dataframe yaitu matriks hanya bisa menyimpan tipe data yang sama sedangkan dataframe boleh memiliki tipe data yang beragam antar data di kolom dan baris. Sebagai contoh dataframe yang terdapat bahasa R adalah *mtcars.*



```
R 4.1.1 · ~/
> mtcars
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

Gambar 5.13 Dataframe mtcars

Sederhananya, dataframe berupa tabel yang terdiri dari *observation* (sebagai baris) dan *variable* (sebagai kolom) yang ditunjukkan pada Gambar 5.14.



Gambar 5.14 Dataframe Terstruktur

Misalkan membuat dataframe diberi nama sebagai *datafr* yang berisi dari 4 *observation* dan 2 *variable* , perhatikan bahasa pemrograman R dibawah ini:

```
Y1 <- c(1,2,3,4)
Z1 <- c("a","b","c","d")
datafr <- data.frame(y1 = Y1, z1 = Z1)
datafr
```

```
R 4.1.1 · ~/
> Y1 <- c(1,2,3,4)
> Z1 <- c("a","b","c","d")
> datafr <- data.frame(Y1 = Y1, Z1 = Z1)
> datafr
  Y1 Z1
1  1  a
2  2  b
3  3  c
4  4  d
```

Gambar 5.15 Hasil Eksekusi DataFrame

Terlihat pada Gambar 5.15 bahwa nilai variabel Y1 akan sebagai isi data dari kolom Y1 begitu juga dengan variabel Z1 ke kolom Z1. Saat sedang menghadapi masalah untuk mengetahui atau mengecek ukuran dimensi dataframe menggunakan fungsi *dim* ( ).

```
dim(datafr)
```

Fungsi dim ( ) pun berfungsi juga ke matrix, dimana vektor pada elemen pertama sebagai *observation* dan elemen kedua sebagai *variable*. Pada saat pembacaan menurut Gambar 1.16 bahwa dataframe *datafr* memiliki 4 *observation* dan 2 *variable*.

```
> dim(datafr)
[1] 4 2
```

Gambar 5.16 Hasil Eksekusi Fungsi dim ( )

Pengguna bahasa pemrograman R ingin mengetahui struktur dalam dari *dataframe* maka dapat menggunakan *str* ( ), fungsi ini bertujuan dapat informasi secara detail atau lengkap dari dataframe seperti banyak observation dan variabel, nama variable, tipe variable dan nilai baris pertama.

```
> str(datafr)
'data.frame':   4 obs. of  2 variables:
 $ Y1: num  1 2 3 4
 $ Z1: chr  "a" "b" "c" "d"
```

Gambar 5.17 Fungsi Str ( )

Diketahui bahwa *datafr* merupakan dataframe yang memiliki ukuran 4 *observation* (obs.) dan 2 *variable* (Y1 tipe data *numeric* dan Z1 tipe data *character*). Misalkan, ingin mengambil nilai variabel *datafr* dari dataframe maka dapat menggunakan *index* atau tanda *dollar* ($). Perlu diperhatikan bahwa banyak cara untuk mengambil nilai variabel *dataframe* berikut beberapa contoh:

```
> #dengan tanda $
> datafr$Y1
[1] 1 2 3 4
> #dengan indeks urutan variabel
> datafr[,1]
[1] 1 2 3 4
> #dengan double-bracket urutan variabel
> datafr[[1]]
[1] 1 2 3 4
```

Gambar 5.18 Cara Akses Nilai Variabel

5.3 Fungsi dan Paket Library pada R

Aplikasi pemrograman R dirancang dan dikembangkan untuk membantu pengguna dalam mengolah, analisis dan visualisasi data. Menurut Wickham (2015) bahwa pemrogaman R memiliki banyak paket *library* untuk memanipulasi dan pembuatan fungsi yang tersedia untuk analitik data. Misalkan untuk dapatkan rata-rata, nilai minimum dan nilai maksimum dari vektor numerik dapat menggunakan fungsi *mean* ( ), *min* ( ), *max* ( ).

```
y <- seq(1,100, by=4)
mean (y) #rata-rata vektor y
min (y) #nilai minimum vektor y
max (y) #nilai maksimum vektor y
```

```
> y <- seq(1,100,by=4)
> mean(y)
[1] 49
> min(y)
[1] 1
> max(y)
[1] 97
```

Gambar 5.19 Contoh beberapa fungsi pada R

Variabel y pada Gambar 1.19 menggunakan fungsi *seq* ( ) atau *sequence* yang berfungsi untuk mengurutkan bilangan *numeric*, fungsi diatas akan menjalankan dan mengurutkan bilangan pertambahan 4 karena ada keterangan "*by = 4* ". Fungsi *mean* ( ) menghitung rata-rata variabel y dengan nilai 49, *mean* ( ) hitung nilai minimum variabel y dengan nilai 1 dan *max* ( ) hitung nilai maksimum variabel y dengan nilai 97.

Fungsi yang dijalankan pada bahasa pemrograman R berasal dari *package* atau paket-paket dalam ruang perpustakaan R istilah lainnya adalah *library*. *Package* itu sendiri secara pengertiannya berupa fungsi-fungsi yang memudahkan pengguna dalam hal ini seorang Data Scientist dalam menggunakan bahasa R. Misalkan pengguna akan menjalankan dataset *flights* dari library R maka harus dipastikan paket library tersebut harus terinstall terlebih dahulu di environment R.

```
install.packages("nycflights13")
```

Dataset *flights* sudah berada dalam *package* atau paket *nycflight13*, dapat kita aktifkan dengan *library(nycflights13)*.

```
library(nycflights13)
```

Jalankan fungsi *head*( ) untuk menampilkan beberapa *row* bagian atas dari dataset *flights*.

```
head(flights)
```

Gambar 5.20 Contoh Penggunaan Package dan Library nycflights13

5.4 Logika Loop dan IF pada R

Ketika Data Scientist sedang menuliskan bahasa pemrograman yang pastinya akan menjumpai operator logika, sebagaimana dapat diketahui pada logika komputer hanya bernilai *false* (direpresentasikan angka 0) dan *true* (direpresentasikan angka 1). Berikut beberapa operator dan logika ditunjukkan pada Tabel 5.2.

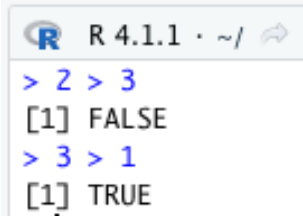
Tabel 5.2 Tabel Operator dan Logika

Logika	**Operator**
Atau	"|"
Dan	"&"

Negasi	“!”
Tidak sama dengan	“!=”
Sama dengan	“==”
Lebih besar	“ > “
Lebih Kecil	“ < “
Lebih besar dari sama dengan	“ >= ”
Lebih kecil dari sama dengan	“ <= ”

Tabel operator logika diatas, salah satunya adalah operator perbandingan yang akan membandingkan nilai atau bilangan *numeric.*

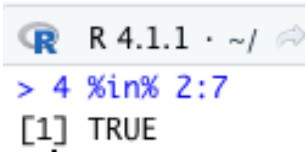
```
2 > 3 #tidak lebih besar dari 3
3 > 1 # lebih besar dari 1
```



Gambar 5.21 Operator Logika Perbandingan

Contoh operator lainnya adalah %in% yang berfungsi untuk mengecek nilai yang berada dalam variabel vektor.

```
4 %in% 2:7
```



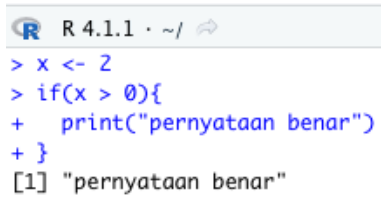
Gambar 5.22. Operator in

Hasil pada bahasa pemrograman R akan mengeluarkan statement TRUE, karena operator %in% berfungsi mengecek nilai *numerik* atau bilangan 4 terdapat dalam variabel *numeric* (2 :7 atau 2, 3, 4, 5, 6, 7). Begitu juga, operator in dapat mengecek tipe data *character*.

Seperti bahasa pemrograman yang lainnya, bahasa R sendiri mendukung fungsi if( ) sebagai bentuk pernyataan pada satu kondisi.

```
x <- 2
if(x > 0){
  print("pernyataan benar")
}
```

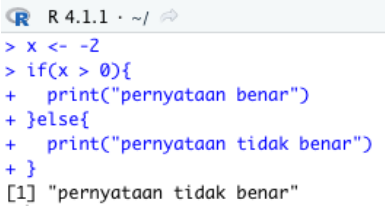
Gambar 1.1 Ditulis Century 12. **Tulis sumber gambar**



Gambar 5.22 Statemen IF di R

Pada Gambar 5.23 bahwa variabel x berisi nilai angka 2, kemudian diberikan statemen kondisi jika x > 0 maka akan mengeluarkan statemen "pernyataan benar". Selain itu, statemen kondisi pada bahasa R memiliki dua pernyataan yang saling tidak berkaitan, berikut contohnya:

```
x <- -2
if(x > 0){
  print("pernyataan benar")
}else{
  print("pernyataan tidak benar")
}
```



Gambar 5.23 Statement IF-ELSE di R

5.5 *Import dan Cleaning* Data

Seorang data scientist sebelum memulai mengolah dan analisis data, dipastikan *dataset* sudah ter-*import* ke dalam pemrograman R, dimana memiliki fungsi untuk membaca file xls atau xlsx, csv, txt dan lain-lain pada suatu *library*. *Library* atau *package tidyverse* digunakan untuk memanggil fungsi-fungsi yang dapat membaca dan import file ke pemrograman R yang berbentuk *tabular* terdiri dari kolom sebagai variabel dan baris sebagai *observation* ke dalam aplikasi *spreadsheet* seperti

google sheets, excel dan lain-lain yang dapat dimanipulasi menggunakan fungsionalitas aplikasi pemrograman R (Joseph, 2022).

```
install.package("tidyverse") #install package
library(tidyverse)

library(readxl) #importing excel files
library(readr) #importing csv
```

Dapat unduh dan gunakan file sampel sederhana pada link dibawah ini untuk di-*import* ke dalam R :

https://drive.google.com/file/d/1PPxxaZbINZ3HJ5qTLgcnnhEf9FGE1ph2/view

Library readr sebagai *package* yang dibutuhkan untuk akses atau membaca fungsi *read_csv* ( ). Perhatikan contoh bahasa R berikut:

```
library(readr)
sampel_csv                                    <-
read_csv("/Users/mackbookair/Downloads/sample_c
sv.csv")
view(sampel_csv)
```

Variabel sampel_csv akan membaca file csv pada tempat penyimpanan file tersebut didalam komputer pengguna. Disini penulis menyimpan file *sample_csv.csv* berada dalam direktori Users/mackbookair/Downloads, file akan masuk atau *import* ke *environment* R artinya file telah siap digunakan untuk proses pengolahan dan analisis data. Pengguna dapat mengecek file csv tersebut dengan menjalankan fungsi *view* ( ) yang akan menampilkan *pop-up* isi dari tabular file *sample_csv.csv*.

	state	capital
1	Michigan	Lansing
2	California	Sacramento
3	New Jersey	Trenton

Gambar 5.24 File *sample_csv.csv*

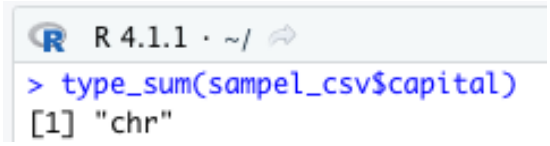
Data Scientist perlu mengetahui tipe data suatu variabel, karena tipe data yang berbeda memiliki perilaku yang berbeda juga, tipe data yang sering dijumpai sebagai berikut: (Joseph, 2022)

Tabel 5.3 Beberapa Tipe Data (Joseph, 2022)

Tipe Data	Singkatan	Objek
Integer	int	Bilangan bulat
Double	dbl	Bilangan real
Character	Chr	Karakter

Pengguna dapat mengecek tipe data dari variabel sampel_csv bagian atribut "capital" di dataset dengan menggunakan fungsi *type_sum* ( ). Perhatikan contoh bahasa R berikut:

```
type_sum(sampel_csv$capital)
```



Gambar 5.25 Fungsi type_sum ( )

Bagi seorang data scientist pekerjaan pembersihan data bisa dibilang proses yang paling panjang. Pembersihan data merupakan proses deteksi dan koreksi *record* data yang rusak atau tidak akurat dari *record* data yang mengacu pada identifikasi bagian data yang tidak lengkap, tidak benar, tidak akurat atau tidak relevan dan kemudan mengganti, memodifikasi atau hapus data yang kotor (Wu,2013). Mari gunakan dataset yang ada dalam library aplikasi R yaitu dataset *World Health Organization Global Tuberculosis Report*, hanya dengan ketikan *who* di bagian conlose R sebagai bahan untuk *cleansing data*.

```
> who
# A tibble: 7,240 × 60
   country     iso2  iso3   year new_sp_m014 new_sp_m1524 new_sp_m2534 new_sp_m3544 new_sp_m4554
   <chr>       <chr> <chr> <int>       <int>        <int>        <int>        <int>        <int>
 1 Afghanistan AF    AFG    1980          NA           NA           NA           NA           NA
 2 Afghanistan AF    AFG    1981          NA           NA           NA           NA           NA
 3 Afghanistan AF    AFG    1982          NA           NA           NA           NA           NA
 4 Afghanistan AF    AFG    1983          NA           NA           NA           NA           NA
 5 Afghanistan AF    AFG    1984          NA           NA           NA           NA           NA
 6 Afghanistan AF    AFG    1985          NA           NA           NA           NA           NA
 7 Afghanistan AF    AFG    1986          NA           NA           NA           NA           NA
 8 Afghanistan AF    AFG    1987          NA           NA           NA           NA           NA
 9 Afghanistan AF    AFG    1988          NA           NA           NA           NA           NA
10 Afghanistan AF    AFG    1989          NA           NA           NA           NA           NA
# … with 7,230 more rows, and 51 more variables: new_sp_m5564 <int>, new_sp_m65 <int>,
#   new_sp_f014 <int>, new_sp_f1524 <int>, new_sp_f2534 <int>, new_sp_f3544 <int>,
#   new_sp_f4554 <int>, new_sp_f5564 <int>, new_sp_f65 <int>, new_sn_m014 <int>,
```

Gambar 5.26 Dataset WHO

Pada dataset *who* yang ditunjukkan pada Gambar 1.27 banyak sekali data yang hilang (*missing*) dan ketidakefisiensian pada dataset tersebut sehingga harus dibuatkan jadi satu kolom pada kasus *new_sp_m014* sampai *newrel_f65* yang diberi kolom baru dengan nama "*cases_type*". Selanjutnya, dataset tersebut dilakukan pembersihan atau membuang data yang hilang (*missing*) dengan fungsi filter(!is.na(column)).

```
#penggabungan dataset new_sp_m14 sampai newrel_f65
who_clean <- who %>%
    select(-iso2,-iso3)%>%
    pivot_longer(cols=new_sp_m-14:newrel_f65,
names_to="cases_type", values_to="cases")
who_clean

#pembersihan missing value (NA)
who_clean_NA <- who_clean%>%
    filter(!is.na(cases))
who clean NA
```

```
> who_clean
# A tibble: 405,440 × 4
   country      year case_type    cases
   <chr>       <int> <chr>        <int>
 1 Afghanistan  1980 new_sp_m014     NA
 2 Afghanistan  1980 new_sp_m1524    NA
 3 Afghanistan  1980 new_sp_m2534    NA
 4 Afghanistan  1980 new_sp_m3544    NA
 5 Afghanistan  1980 new_sp_m4554    NA
 6 Afghanistan  1980 new_sp_m5564    NA
 7 Afghanistan  1980 new_sp_m65      NA
 8 Afghanistan  1980 new_sp_f014     NA
 9 Afghanistan  1980 new_sp_f1524    NA
10 Afghanistan  1980 new_sp_f2534    NA
# … with 405,430 more rows
```

Gambar 5.27 Penggabungan Data dengan Pivot

Penggabungan data dengan fungsi pivot berhasil dilakukan, data telah tersusun dengan rapih dengan jumlah baris data (*row*) sebanyak 405.430, data variabel who_clean masih banyak data yang hilang (*missing value*).

```
> who_clean_NA
# A tibble: 76,046 × 4
   country      year case_type    cases
   <chr>       <int> <chr>        <int>
 1 Afghanistan  1997 new_sp_m014      0
 2 Afghanistan  1997 new_sp_m1524    10
 3 Afghanistan  1997 new_sp_m2534     6
 4 Afghanistan  1997 new_sp_m3544     3
 5 Afghanistan  1997 new_sp_m4554     5
 6 Afghanistan  1997 new_sp_m5564     2
 7 Afghanistan  1997 new_sp_m65       0
 8 Afghanistan  1997 new_sp_f014      5
 9 Afghanistan  1997 new_sp_f1524    38
10 Afghanistan  1997 new_sp_f2534    36
# … with 76,036 more rows
```

Gambar 5.28 Hasil Pembersihan Missing Data

Pembersihan data dengan fungsi filter(!is.na(column)) berhasil dilakukan, terlihat sekali jumlah baris data (row) sebelum dan sesudah. Data yang sudah dibersihkan (405.430 (Gambar 1.28) dikurangi 76.036 (Gambar 1.29)) adalah 329.394 *rows* data.

5.6 Visualisasi Data

Hasil analisa perlu direpresentasikan dengan baik agar dapat dibaca oleh pengguna atau organisasi, maka visualisasi data berperan penting dalam mengkemas informasi dan komunikasi dari hasil olah analisa data. Hal tersebut sangat membantu pengguna atau organisasi untuk membuat keputusan analisa yang akan diambil dan diterapkan. Pada *environment* bahasa pemrograman R memiliki *library* untuk memvisualisasikan data yang beragam, umumnya *library* yang sering digunakan untuk visualisasi data yaitu ggplot ( ) dan lattice ( ).

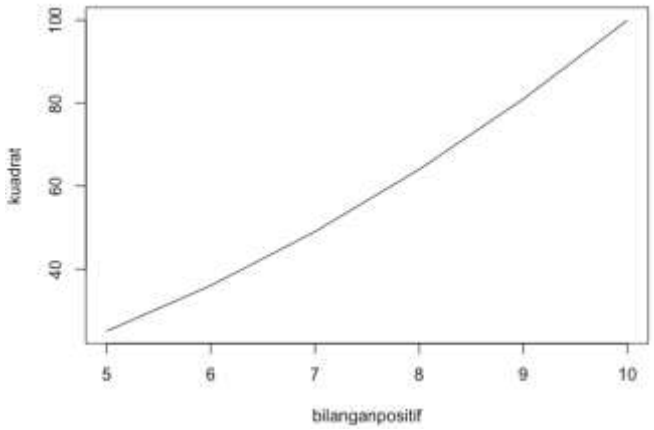
Penulis disini menggunakan beberapa fungsi visualisasi dari *library* R yiatu fungsi plot ( ) dengan format, plot(x, y, type=”p”). Agar penjelasan visualisasi data, penulis sajikan contoh penggunaan grafik dari fungsi plot( ).

```
#membuat variabel bilanganpositif
bilanganpositif <- c(5:10)
kuadrat <- bilanganpositif^2

#membuat grafikkuadrat
grafikkuadrat <- plot(bilanganpositif, kuadrat,
type="l")
```

```
> bilanganpositif <- c(5:10)
> kuadrat <- bilanganpositif^2
> grafikkuadrat <- plot(bilanganpositif, kuadrat, type="l")
```

Gambar 5.29 Fungsi Plot( )

Pada Gambar 5.30 bahwa variabel bilanganpositif memiliki nilai vektor dari angka 5 sampai 10 dan variabel kuadrat akan mengkuadratkan nilai-nilai dari variabel bilanganpositif. Agar hasil perhitungan kuadrat dapat dilihat dengan jelas dan mudah dibaca dengan grafik maka dapat menggunakan dan menjalankan fungsi plot( ). Berikut hasil eksekusi *coding* di atas.



Gambar 5.30 Grafik Hasil Kuadrat Bilanganpositif

5.7 Penutup

Aplikasi pemrograman R sangat membantu akademisi, praktisi dan professional untuk mengolah data dan analisa data yang berguna untuk mendukung suatu keputusan dari organisasi pengguna tersebut. *Envrionment* pemrograman R didukung dengan fasilitas kamus atau perpustakaan R yang sering diistilahkan sebagai *library* sehingga pengguna harus mengetahui fungsi *library* yang akan digunakan untuk memecahkan permasalahan yang sedang dikerjakan.

Seorang data scientist harus memahami cara kerja analisa data yang dimulai dari ekstraksi (unduh), import data, pembersihan data (*cleaning*), pengolahan sederhana dengan rumus statistika yang umum sering digunakan seperti *mean*, *modus* dan lain-lain, lalu divisualisasikan dengan *library* ggplot( ) dan sebagainya. Penulisan chapter ini memperkenalkan secara fundamental pemrograman bahasa R hingga praktek cara membuat bahasa (*coding*) R yang sederhana bahwasannya aplikasi pemrograman R tidak kalah hebatnya dengan aplikasi pengolah data yang semacamnya seperti SPSS, Excel, dan sebagainya.

Bab 6

Python For Data Scientist

6.1 Pengantar

Python merupakan salah satu bahasa pemrograman *open source* yang digunakan oleh para ilmuan data (*data scientist*) yang dapat digunakan untuk membersihkan data, membuat visualisasi, dan membangun model. Bahasa pemrograman *Python* bisa ditafsirkan, dan merupakan bahasa tingkat tinggi yang memungkinkan pendekatan yang lebih baik untuk pemrograman berorientasi objek. *Python* merupakan salah satu bahasa terbaik yang digunakan oleh ilmuwan data untuk berbagai proyek atau aplikasi ilmu data. Fungsionalitas yang diperlukan untuk menangani matematika, statistik, dan fungsi ilmiah disediakan dalam Bahasa pemrograman *Python*. Kemudahan penggunaan dan kesederhanaan sintaksis (tata kalimat bahasa yang membentuk sebuah kalimat) menjadi alasan utama *Python* banyak digunakan dalam *data science.*

6.2 Modelling

Model *Parsimonious* merupakan model yang mencapai tingkat prediksi yang diinginkan dengan variabel prediktor sesedikit mungkin.

6.2.1 Variabel

Variabel dapat dibedakan atas 2 jenis, yaitu:

X = Variabel Independen (prediktor)

Y = Variabel dependen (target)

6.2.2 Tipe Data

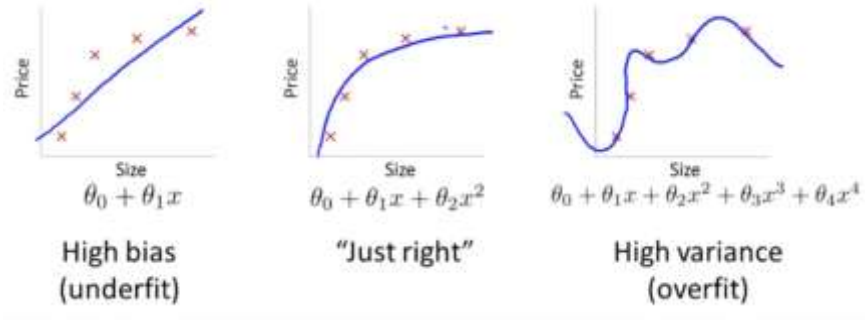
Jenis data sangat penting karena menentukan jenis tes apa yang dapat diterapkan padanya. Terdapat 2 jenis tipe data yaitu:

a. *Continuous*: Juga dikenal sebagai kuantitatif. Jumlah nilai yang tidak terbatas.

b. *Categorical*: Juga dikenal sebagai diskrit atau kualitatif. Memperbaiki jumlah nilai atau kategori.

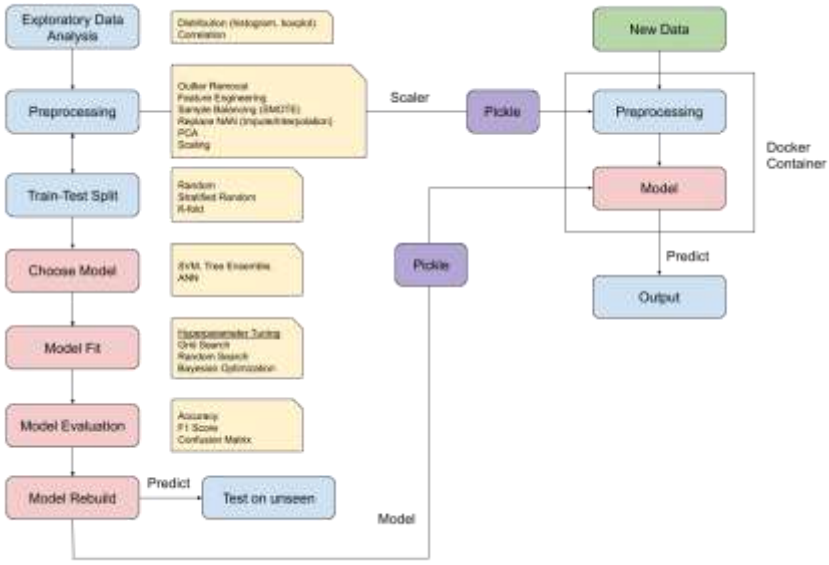
6.2.3 *Bias-Variance Tradeoff*

Algoritma prediksi terbaik adalah algoritma yang memiliki Kemampuan Generalisasi yang baik. Algoritma akan dapat memberikan prediksi yang akurat terhadap data baru dan data yang sebelumnya tidak terlihat. Bias Tinggi merupakan hasil dari *Underfitting* model. Ini biasanya merupakan hasil dari asumsi yang salah, dan menyebabkan model menjadi terlalu umum. Varians Tinggi merupakan hasil dari *Overfitting* model, dan ini akan memprediksi dataset pelatihan dengan sangat akurat, tetapi tidak dengan dataset baru yang tidak terlihat. Model terbaik dengan akurasi tertinggi merupakan jalan tengah di antara keduanya.



Gambar 6.1 Gambaran Model Prediktif (sumber: Andrew Ng's lecture)

6.2.4 Langkah-langkah Membangun Model Prediktif



Gambar 6.2 Arsitektur untuk pembuatan model untuk klasifikasi terawasi (sumber: Dokumentasi Data Science)

6.2.4.1 Seleksi Fitur, Preprocessing, Ekstraksi

Berikut merupakan langkah dalam seleksi fitur, preprocessing dan ekstraksi:

a. Hapus fitur yang memiliki NAN terlalu banyak atau isi NAN dengan nilai lain

b. Hapus fitur yang akan menyebabkan kebocoran data

c. Mengkodekan fitur kategorikal menjadi bilangan bulat

d. Ekstrak fitur baru yang berguna (antara dan di dalam fitur saat ini)

6.2.4.2. Normalisasi Fitur

Dengan pengecualian model Tree dan Naive Bayes, teknik pembelajaran mesin lainnya seperti Neural Networks, KNN, SVM harus memiliki skala fiturnya.

6.2.4.3. *Train Test Split*

Pisahkan set data menjadi set data Latih dan Uji. Secara default, sklearn menetapkan 75% untuk melatih & 25% untuk menguji secara acak. Keadaan acak (seed) dapat dipilih untuk memperbaiki pengacakan.

```
from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test
= train_test_split(predictor, target, test_size=0.25,
random_state=0)
```

6.2.4.4. Pembuatan Model

Pilih model dan atur parameter model (jika ada).

```
clf = DecisionTreeClassifier()
```

6.2.4.5. *Fit Model*

Sesuaikan model menggunakan set data latih.

```
model = clf.fit(X_train, y_train)


>>> print model
DecisionTreeClassifier(class_weight=None, criterion='gini',
max_depth=None,
        max_features=None, max_leaf_nodes=None,
min_samples_leaf=1,
        min_samples_split=2, min_weight_fraction_leaf=0.0,
        presort=False, random_state=None, splitter='best')
```

6.2.4.6. Model Uji

Uji model dengan memprediksi identitas data yang tidak terlihat menggunakan dataset uji.

```
y_predict = model.predict(X_test)
```

6.2.4.7. Score Model

Menggunakan confusion matrix, persentase akurasi, dan skor f1 untuk mendapatkan akurasi prediksi.

```
>>> print sklearn.metrics.confusion_matrix(y_test,
predictions)
[[14  0  0]
 [ 0 13  0]
 [ 0  1 10]]


import sklearn.metrics
print sklearn.metrics.accuracy_score(y_test, y_predict)*100,
'%'
>>> 97.3684210526 %
```

6.2.4.8. Validasi silang

Ketika semua kode berfungsi dengan baik, hapus bagian train-test dan gunakan Grid Search Cross Validation untuk menghitung parameter terbaik dengan validasi silang.

6.2.4.9. Model Akhir

Terakhir, buat ulang model menggunakan kumpulan data lengkap, dan parameter yang dipilih diuji.

6.2.5. Analisis Cepat untuk Multi-Model

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

from sklearn.svm import LinearSVC
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from xgboost import XGBClassifier

from sklearn.metrics import accuracy_score, f1_score
from statistics import mean
import seaborn as sns

# models to test
svml = LinearSVC()
svm = SVC()
rf = RandomForestClassifier()
xg = XGBClassifier()
xr = ExtraTreesClassifier()
# iterations
classifiers = [svml, svm, rf, xr, xg]
names = ['Linear SVM', 'RBF SVM', 'Random Forest', 'Extremely
Randomized Trees', 'XGBoost']
```

```
results = []


# train-test split
X = df[df.columns[:-1]]
# normalise data for SVM
X = StandardScaler().fit(X).transform(X)
y = df['label']
X_train, X_test, y_train, y_test = train_test_split(X, y,
random_state=0)


for name, clf in zip(names, classifiers):
    model = clf.fit(X_train, y_train)
    y_predict = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_predict)
    f1 = mean(f1_score(y_test, y_predict, average=None))
    results.append([fault, name, accuracy, f1])
```

Untuk membandingkan hasilnya

```
final = pd.DataFrame(results, columns=['Fault
Type','Model','Accuracy','F1 Score'])
final.style.background_gradient(cmap='Greens')
```

	Model	Accuracy	F1 Score
0	Linear SVM	0.84	0.832134
1	RBF SVM	0.687692	0.66756
2	Random Forest	0.912308	0.910632
3	Extremely Randomized Trees	0.870769	0.868625
4	XGBoost	0.938462	0.93719

Gambar 6.3 Hasil Analisis Cepat

6.3 Learning

6.3.1 Dataset

Terdapat kumpulan data bawaan yang disediakan dalam paket statsmodels dan sklearn.

6.3.1.1 Model statistik

Dalam statsmodels, banyak dataset R dapat diperoleh dari fungsi sm.datasets.get_rdataset(). Untuk melihat deskripsi setiap set data, gunakan print(duncan_prestige.__doc__).

(Sumber: https://www.statsmodels.org/devel/datasets/index.html)

```
import statsmodels.api as sm
prestige = sm.datasets.get_rdataset("Duncan", "car", cache=True).data
print prestige.head()
type income education prestige
accountant prof 62 86 82
pilot prof 72 76 83
architect prof 75 92 90
author prof 55 90 76
chemist prof 64 86 90
```

6.3.1.2 Sklearn

Ada lima kumpulan data umum di sini. Untuk data yang lain, silahkan lihat http://scikit-learn.org/stable/datasets/index.html. Untuk melihat deskripsi setiap dataset, gunakan print boston['DESCR'].

load_boston([return_X_y])	Memuat dan mengembalikan dataset harga rumah boston (regresi).
load_iris([return_X_y])	Memuat dan mengembalikan dataset iris (klasifikasi).
load_diabetes([return_X_y])	Memuat dan mengembalikan dataset diabetes (regresi).
load_digits([n_class, return_X_y])	Memuat dan mengembalikan dataset digit (klasifikasi).
load_linnerud([return_X_y])	Memuat dan mengembalikan dataset linnerud (regresi multivariat).

```
from sklearn.datasets import load_iris

# Load Iris data (https://en.wikipedia.org/wiki/Iris_flower_data_set)

iris = load_iris()

# Load iris into a dataframe and set the field names

df = pd.DataFrame(iris['data'], columns=iris['feature_names'])

df.head()
```

```
sepal length (cm)  sepal width (cm)  petal length (cm)  petal
width (cm)

0           5.1             3.5              1.4              0.2

1           4.9             3.0              1.4              0.2

2           4.7             3.2              1.3              0.2

3           4.6             3.1              1.5              0.2

4           5.0             3.6              1.4              0.2

# Feature names are in .target & .target_names

>>> print iris.target_names[:5]

>>> ['setosa' 'versicolor' 'virginica']

>>> print iris.target

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]

# Change target to target_names & merge with main dataframe

df['species'] = pd.Categorical.from_codes(iris.target,
iris.target_names)

print df['species'].head()

sepal length (cm)  sepal width (cm)  petal length (cm)  petal
width (cm)

0           5.1             3.5              1.4              0.2

1           4.9             3.0              1.4              0.2

2           4.7             3.2              1.3              0.2

3           4.6             3.1              1.5              0.2
```

```
4          5.0          3.6          1.4          0.2
0    setosa
1    setosa
2    setosa
3    setosa
4    setosa
Name: species, dtype: category
Categories (3, object): [setosa, versicolor, virginica]
```

6.3.1.3 Vega-Datasets

Tidak built-in tetapi dapat diinstal melalui pip install vega_datasets. Selengkap dapat dilihat di https://github.com/jakevdp/vega_datasets.

```
from vega_datasets import data
df = data.iris()
df.head()
```

	petalLength	petalWidth	sepalLength	sepalWidth	species
0	1.4	0.2	5.1	3.5	setosa
1	1.4	0.2	4.9	3.0	setosa
2	1.3	0.2	4.7	3.2	setosa
3	1.5	0.2	4.6	3.1	setosa
4	1.4	0.2	5.0	3.6	setosa

Untuk membuat daftar semua dataset, gunakan list_datasets()

```
>>> data.list_datasets()
['7zip', 'airports', 'anscombe', 'barley', 'birdstrikes', 'budget', \
 'budgets', 'burtin', 'cars', 'climate', 'co2-concentration', 'countries', \
 'crimea', 'disasters', 'driving', 'earthquakes', 'ffox', 'flare', \
 'flare-dependencies', 'flights-10k', 'flights-200k', 'flights-20k', \
 'flights-2k', 'flights-3m', 'flights-5k', 'flights-airport', 'gapminder', \
 'gapminder-health-income', 'gimp', 'github', 'graticule', 'income', 'iris', \
 'jobs', 'londonBoroughs', 'londonCentroids', 'londonTubeLines', 'lookup_groups', \
 'lookup_people', 'miserables', 'monarchs', 'movies', 'normal-2d', 'obesity', \
 'points', 'population', 'population_engineers_hurricanes', 'seattle-temps', \
 'seattle-weather', 'sf-temps', 'sp500', 'stocks', 'udistrict', 'unemployment', \
 'unemployment-across-industries', 'us-10m', 'us-employment', 'us-state-capitals', \
 'weather', 'weball26', 'wheat', 'world-110m', 'zipcodes']
```

6.4 Analisis Eksplorasi

Analisis data eksplorasi atau *Exploratory Data Analysis* (EDA) merupakan langkah penting untuk memahami data dengan lebih baik, yaitu untuk merekayasa dan memilih fitur sebelum pemodelan. EDA sering membutuhkan keterampilan

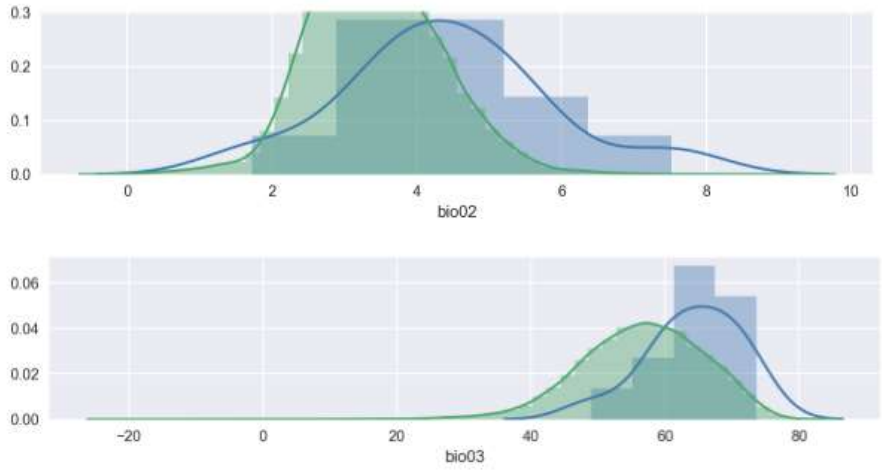
dalam visualisasi untuk menginterpretasikan data dengan lebih baik.

6.4.1 Univariat

6.4.1.1 Plot Distribusi

Saat memplot distribusi, penting untuk membandingkan distribusi data set dan data uji. Jika set tes sangat spesifik untuk fitur tertentu, model akan kurang fit dan memiliki akurasi yang rendah.

```
import seaborn as sns
import matplotlib.pyplot as plt
%config InlineBackend.figure_format = 'retina'
%matplotlib inline


for i in X.columns:
  plt.figure(figsize=(15,5))
  sns.distplot(X[i])
  sns.distplot(pred[i])
```



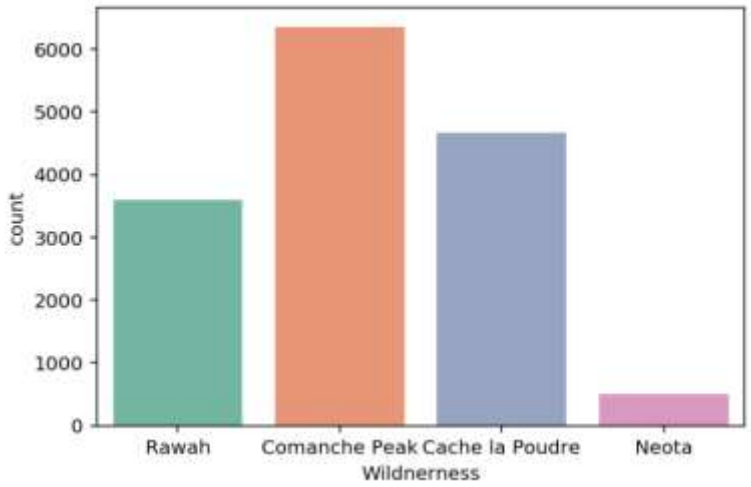
Gambar 6.4 Visualisasi Plot

6.4.1.2 Hitung Plot

Untuk fitur kategori, jika ingin melihat apakah mereka memiliki ukuran sampel yang cukup untuk setiap kategori.

```
import seaborn as sns
import matplotlib.pyplot as plt
%config InlineBackend.figure_format = 'retina'
%matplotlib inline

df['Wildnerness'].value_counts()

Comanche Peak      6349
Cache la Poudre    4675
Rawah              3597
Neota              499
Name: Wildnerness, dtype: int64
```

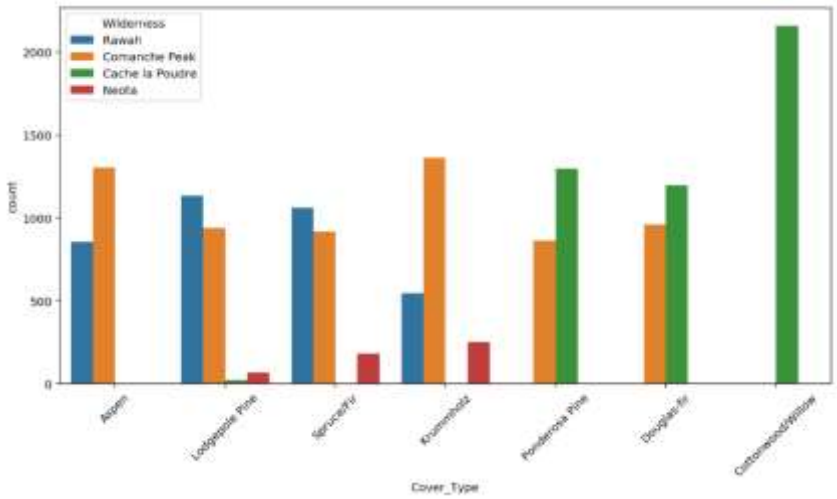
```
cmap = sns.color_palette("Set2")
sns.countplot(x='Wildnerness',data=df, palette=cmap);
plt.xticks(rotation=45);
```



Gambar 6.5 Visualisasi Hasil Plot

Untuk memeriksa kemungkinan hubungan dengan target, bisa dengan menempatkan fitur di bawah *Hue*.

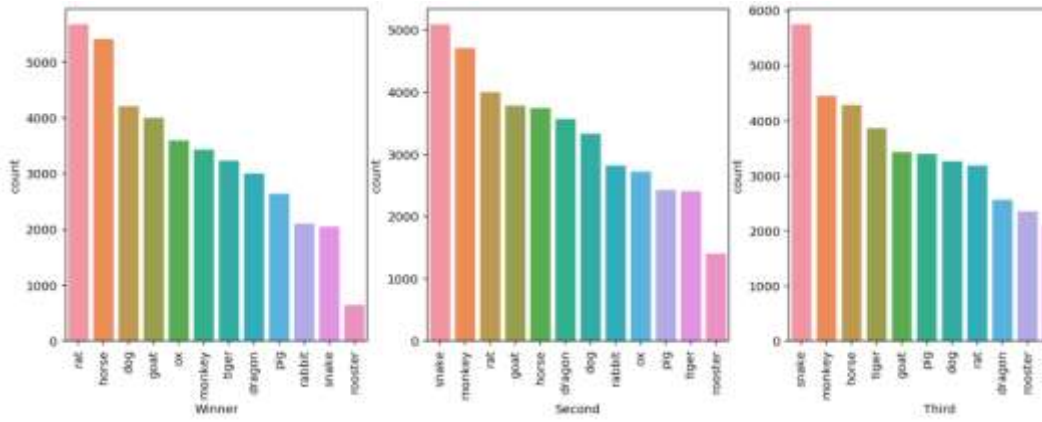
```
plt.figure(figsize=(12,6))
sns.countplot(x='Cover_Type',data=wild, hue='Wilderness');
plt.xticks(rotation=45);
```



Gambar 6.6 Visualisasi Hubungan Target

Multiple Plots

```
fig, axes = plt.subplots(ncols=3, nrows=1, figsize=(15, 5)) # note only for 1 row or 1 col, else need to flatten nested list in axes

col = ['Winner','Second','Third']


for cnt, ax in enumerate(axes):

    sns.countplot(x=col[cnt], data=df2, ax=ax, order=df2[col[cnt]].value_counts().index);


for ax in fig.axes:

    plt.sca(ax)

    plt.xticks(rotation=90)
```

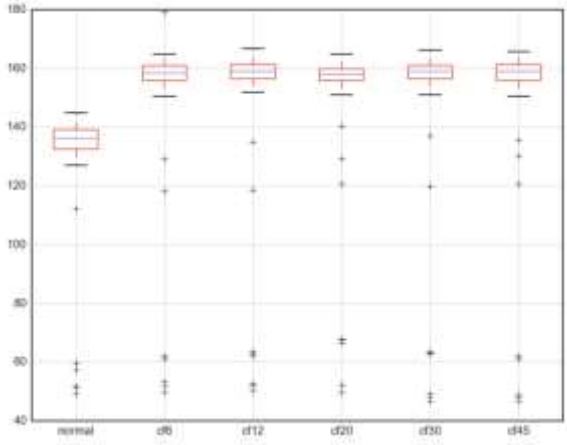


Gambar 6.7 Visualisasi Multiple Plot

6.4.1.3 Box Plots

Pada Box Plot, menggunakan data 50 persen untuk membandingkan data di antara kelas yang berbeda, cukup mudah untuk menemukan fitur yang memiliki kepentingan prediksi tinggi jika tidak tumpang tindih. Juga dapat digunakan untuk deteksi outlier. Fitur harus berkelanjutan. Dari kerangka data yang berbeda, menampilkan fitur yang sama.

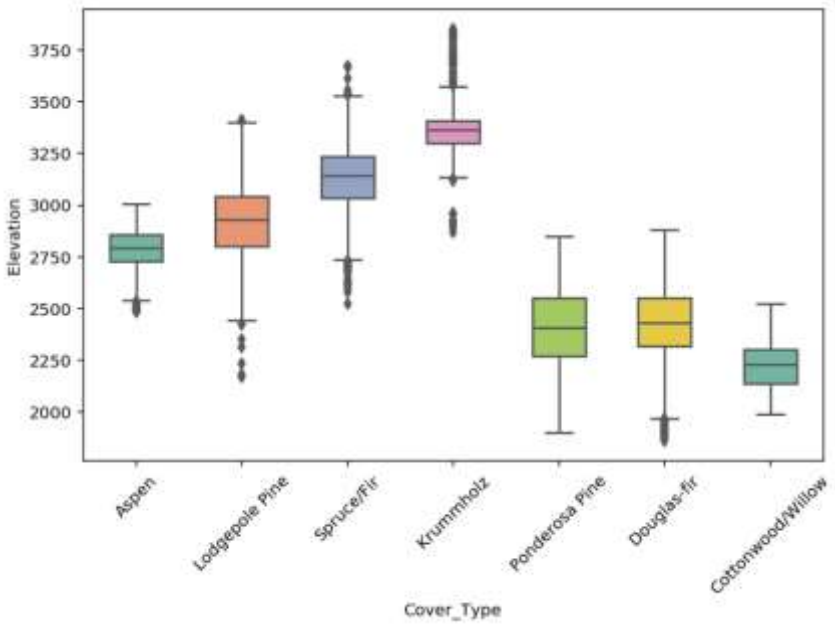
```
df = pd.DataFrame({'normal': normal['Pressure'], 's1':
cf6['Pressure'], 's2': cf12['Pressure'],
                    's3': cf20['Pressure'], 's4': cf30['Pressure'],'s5':
cf45['Pressure']})
df.boxplot(figsize=(10,5));
```



Gambar 6.8 Visualisasi Box Plot

Dari kerangka data yang sama dengan fitur yang dipisahkan oleh label-y yang berbeda.

```
plt.figure(figsize=(7, 5))

cmap = sns.color_palette("Set3")

sns.boxplot(x='Cover_Type', y='Elevation', data=df,
palette=cmap);

plt.xticks(rotation=45);
```

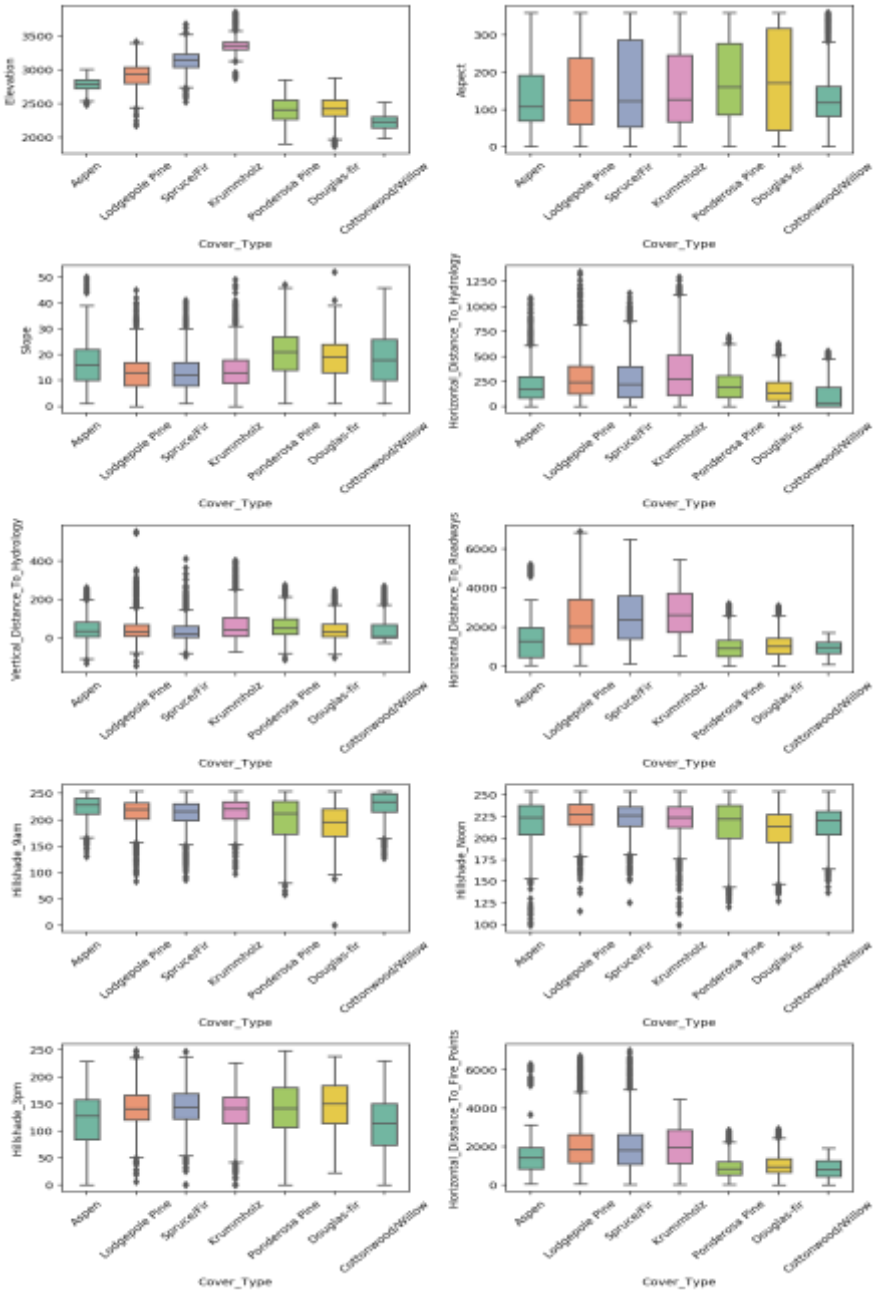


Gambar 6.9 Box Plot Pemisahan oleh label-y

Multiple Plots

```
cmap = sns.color_palette("Set2")


fig, axes = plt.subplots(ncols=2, nrows=5, figsize=(10, 18))
a = [i for i in axes for i in i] # axes is nested if >1 row & >1 col, need to flatten
for i, ax in enumerate(a):
    sns.boxplot(x='Cover_Type', y=eda2.columns[i], data=eda, palette=cmap, width=0.5, ax=ax);


# rotate x-axis for every single plot
for ax in fig.axes:
    plt.sca(ax)
```

```
    plt.xticks(rotation=45)


# set spacing for every subplot, else x-axis will be covered
plt.tight_layout()
```



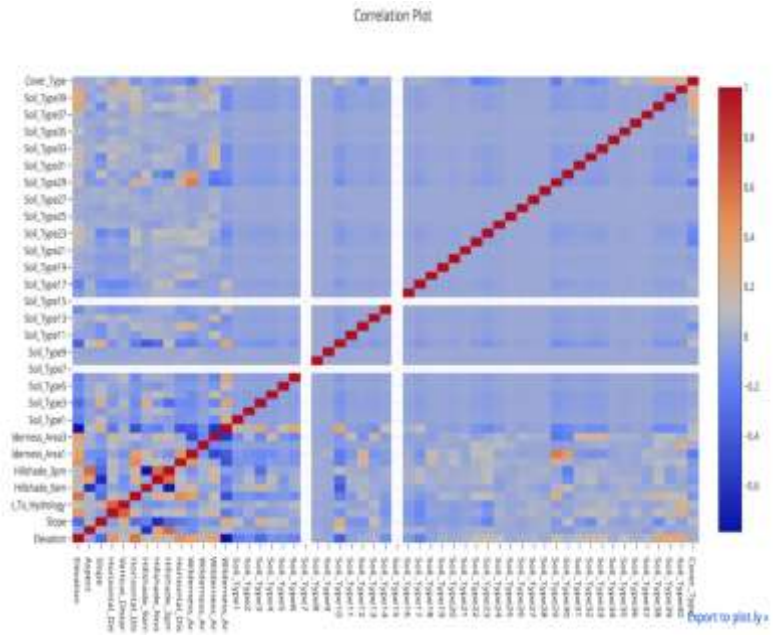
Gambar 6.10 Multiple Box Plot

6.4.2 Multi-Variate

6.4.2.1 Correlation Plots

Heatmaps menunjukkan korelasi keseluruhan yang cepat antar fitur.

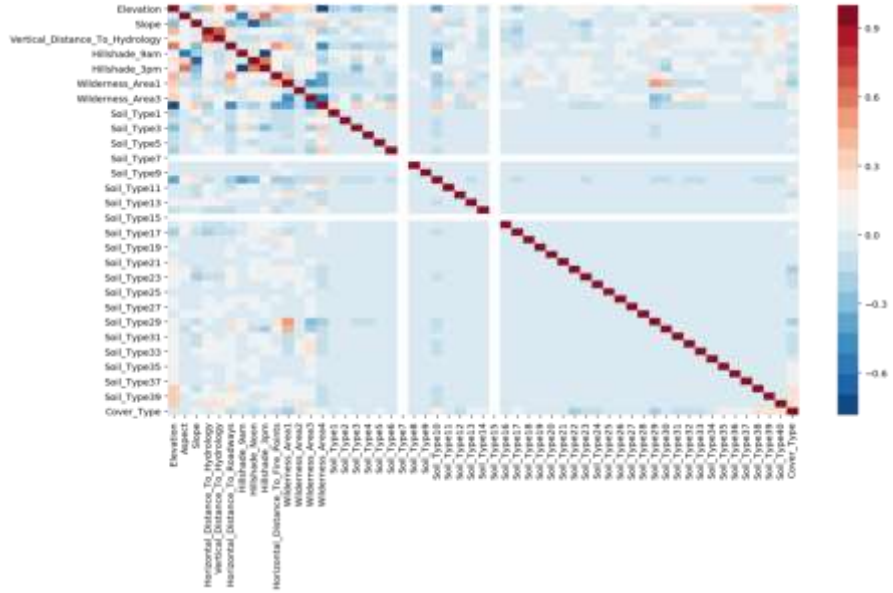
```
from plotly.offline import iplot
from plotly.offline import init_notebook_mode
import plotly.graph_objs as go
init_notebook_mode(connected=True)

# create correlation in dataframe
corr = df[df.columns[1:]].corr()

layout = go.Layout(width=1000, height=600, \
            title='Correlation Plot', \
            font=dict(size=10))
data = go.Heatmap(z=corr.values, x=corr.columns,
y=corr.columns)
fig = go.Figure(data=[data], layout=layout)
iplot(fig)
```



Gambar 6.11 Visualisasi Plot Korelasi

Menggunakan Seaborn

```
import seaborn as sns
import matplotlib.pyplot as plt
%config InlineBackend.figure_format = 'retina'
%matplotlib inline


# create correlation in dataframe
corr = df[df.columns[1:]].corr()


plt.figure(figsize=(15, 8))
sns.heatmap(corr, cmap=sns.color_palette("RdBu_r", 20));
```



Gambar 6.12 Visualisasi menggunakan seaborn

6.5 Penutup

Dalam bab ini, telah dibahas teknik modelling, learning dan visualisasi yang sering digunakan oleh ilmuan data, termasuk jenis data (numerik dan kategorikal) yang mungkin ditemui dan cara mengategorikannya serta cara memperlakukannya secara berbeda bergantung pada jenis data yang dihadapi.

Bab 7

Data Penginderaan Jauh Satelit

7.1 Pengantar

Penginderaan jauh adalah istilah serapan dari Bahasa Inggris "remote sensing" yang berarti pengambilan atu pengontrolan suatu bidang tertentu untuk lebih pada titik fokus yang dituju. Dalam kajian geografi, istilah kata ini juga dikenal dengan inderaja..

Perkembangan penginderaan jauh ini semakin cepat seiring dengan kemajuan teknologi dirgantara. Sebelumnya penginderaan jauh lebih banyak menggunakan pesawat udara dan balon udara dalam perekaman data permukaan bumi, tetapi seiring dengan perkembangan penerbangan antariksa dan penggunaan satelit untuk berbagai kepentingan termasuk didalamnya perekaman permukaan bumi, maka penginderaan jauh tumbuh berkembang semakin cepat. Demikian pula halnya dengan penggunaan sensor yang di bawa oleh berbagai wahana juga mengalami peningkatan baik dalam jenis sensor yang digunakan maupun tingkat kedetailan hasil penginderaan.

7.2 Pengertian Penginderaan Jauh Menurut Para Ahli

1. American Society of Photogrammetry (1983)

   Penginderaan jauh merupakan pengukuran atau perolehan informasi dari beberapa sifat objek atau fenomena dengan

menggunakan alat perekam yang secara fisik tidak terjadi kontak langsung dengan objek atau fenomena yang dikaji.

2. Avery (1985)

   Penginderaan jauh merupakan upaya untuk memperoleh, menunjukkan (mengidentifikasi), dan menganalisis objek dengan sensor pada posisi pengamatan daerah kajian.

3. Campbell

   Penginderaan jauh adalah ilmu untuk mendapatkan informasi mengenai permukaan bumi, seperti lahan dan air, dari citra yang diperoleh dari jarak jauh.

4. Colwell (1984)

   Penginderaan jauh adalah suatu pengukuran atau perolehan data pada objek di permukaan bumi dari satelit atau instrumen lain di atas atau jauh dari objek yang diindera.

5. Curran (1985)

   Penginderaan jauh adalah penggunaan sensor radiasi elektromagnetik untuk merekam gambar lingkungan bumi yang dapat diinterpretasikan sehingga menghasilkan informasi yang berguna.

6. Lillesand dan Kiefer (1979) dan (2007)

   Penginderaan jauh adalah ilmu dan seni untuk memperoleh informasi tentang objek, wilayah, atau gejala dengan cara menganalisis data yang diperoleh dengan menggunakan alat tanpa kontak langsung terhadap objek, wilayah, atau gejala yang dikaji.

7. Lindgren

   Penginderaan jauh adalah berbagai teknik yang dikembangkan untuk perolehan dan analisis informasi tentang bumi.

8. Welson Dan Bufon

   Penginderaan jauh adalah sebagai suatu ilmu, seni, dan teknik untuk memperoleh objek, area, dan gejala dengan menggunakan alat dan tanpa kontak langsung dengan objek, area, dan gejala tersebut.

7.3 Sejarah Penginderaan Jauh

Tabel 7. 1 Sejarah Penginderaan Jauh

Waktu	**Tokoh / Ilmuwan**	**Keterangan Perkembangan Penginderaan Jauh**
1858	Gaspard Felix Tournachon	Menggunakan balon udara untuk memotret daerah Bievre-Perancis pada ketinggian 180 meter.
1860	James Wallace B	Menggunakan balon udara untuk memotret kota Boston
18 April 1906	G.R. Lawrence	Menggunakan laying-layang dengan tinggi 600 meter untuk memotret bencana gempa dan kebakaran di kota San Fransisco kamera menghasilkan foto negatif berukuran 1.4 x 2.4 meter.
1895	Kolonel Laussedat	Menggunakan layang-layang dan balon udara untuk memetakan suatu wilayah dengan menggunakan fotogrametri.
1882		Penggunaan foto udara untuk analisa pertahanan musuh untuk pada saat perang sipil di Amerika.

1886	Kapten Deville	Menggunakan foto udara untuk membuat foto udara di Kanada.
1902	Wright Bersaudara	Menemukan pesawat udara sehingga wahana yang digunakan tidak lagi menggunakan balon udara. Percobaan pertama kali tahun 1909 di Italia.
1929 - 1939		Perang Dunia I dan II foto udara berkembang pesat untuk pembuatan peta topografi.
1960		Perkembangan Inderaja dengan satelit Dengan diluncurkan satelit, inderaja juga mengalami perkembangan tentang wahana pembawa sensor yaitu dengan menggunakan satelit. Penggunaanya masih sebatas untuk kepentingan militer.

7.4 Komponen-komponen Penginderaan Jauh

a. Sumber Tenaga

Sumber tenaga dalam proses inderaja terdiri atas :

- Sistem pasif adalah sistem yang menggunakan sinar matahari
- Sistem aktif adalah sistem yang menggunakan tenaga buatan seperti gelombang mikro

Jumlah tenaga yang diterima oleh obyek di setiap tempat berbeda-beda, hal ini dipengaruhi oleh beberapa faktor, antara lain :

- Waktu penyinaran

  Jumlah energi yang diterima oleh objek pada saat matahari tegak lurus (siang hari) lebih besar daripada saat posisi miring (sore hari). Makin banyak energi yang diterima objek, makin cerah warna obyek tersebut.

- Bentuk permukaan bumi

  Permukaan bumi yang bertopografi halus dan memiliki warna cerah pada permukaannya lebih banyak memantulkan sinar matahari dibandingkan permukaan yang bertopografi kasar dan berwarna gelap. Sehingga daerah bertopografi halus dan cerah terlihat lebih terang dan jelas.

- Keadaan cuaca

  Kondisi cuaca pada saat pemotretan mempengaruhi kemampuan sumber tenaga dalam memancarkan dan memantulkan. Misalnya kondisi udara yang berkabut menyebabkan hasil inderaja menjadi tidak begitu jelas atau bahkan tidak terlihat.

b. Atmosfer

Lapisan udara yang terdiri atas berbagai jenis gas, seperti O2, CO2, nitrogen, hidrogen dan helium. Molekul-molekul gas yang terdapat di dalam atmosfer tersebut dapat menyerap, memantulkan dan melewatkan radiasi elektromagnetik.

Di dalam inderaja terdapat istilah Jendela Atmosfer, yaitu bagian spektrum elektromagnetik yang dapat mencapai bumi. Keadaan di atmosfer dapat menjadi penghalang pancaran sumber tenaga yang mencapai ke

permukaan bumi. Kondisi cuaca yang berawan menyebabkan sumber tenaga tidak dapat mencapai permukaan bumi.

c. Interaksi antara tenaga dan objek

Interaksi antara tenaga dan obyek dapat dilihat dari rona yang dihasilkan oleh foto udara. Tiap-tiap obyek memiliki karakterisitik yang berbeda dalam memantulkan atau memancarkan tenaga ke sensor.

Objek yang mempunyai daya pantul tinggi akan terilhat cerah pada citra, sedangkan obyek yang daya pantulnya rendah akan terlihat gelap pada citra. Contoh: Permukaan puncak gunung yang tertutup oleh salju mempunyai daya pantul tinggi yang terlihat lebih cerah, daripada permukaan puncak gunung yang tertutup oleh lahar dingin.

d. Sensor Dan Wahana

- **Sensor**

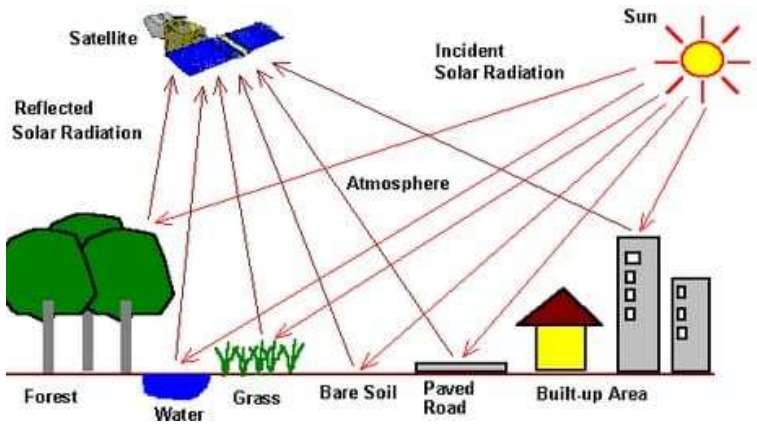
Merupakan alat pemantau yang dipasang pada wahana, baik pesawat maupun satelit. Sensor dapat dibedakan menjadi dua :

1. **Sensor fotografik**, merekam obyek melalui proses kimiawi. Sensor ini menghasilkan foto. Sensor yang dipasang pada pesawat menghasilkan citra foto (foto udara), sensor yang dipasang pada satelit menghasilkan citra satelit (foto satelit)
2. **Sensor elektronik**, bekerja secara elektrik dalam bentuk sinyal. Sinyal elektrik ini direkam dalam pada pita magnetik yang kemudian dapat diproses menjadi data visual atau data digital dengan menggunakan komputer. Kemudian lebih dikenal dengan sebutan citra.

- **Wahana**

  Adalah kendaraan/media yang digunakan untuk membawa sensor guna mendapatkan inderaja. Berdasarkan ketinggian persedaran dan tempat pemantauannya di angkasa, wahana dapat dibedakan menjadi tiga kelompok:

  1. **Pesawat terbang rendah** sampai menengah yang ketinggian peredarannya antara 1.000 – 9.000 meter di atas permukaan bumi
  2. **Pesawat terbang tinggi**, yaitu pesawat yang ketinggian peredarannya lebih dari 18.000 meter di atas permukaan bumi
  3. **Satelit**, wahana yang peredarannya antara 400 km – 900 km diluar atmosfer bumi.



e. Perolehan Data

Data yang diperoleh dari inderaja ada 2 jenis :

- **Data manual**, didapatkan melalui kegiatan interpretasi citra. Guna melakukan interpretasi citra secara manual diperlukan alat bantu

bernamastereoskop. Stereoskop dapat digunakan untuk melihat objek dalam bentuk tiga dimensi.

- **Data numerik** (digital), diperoleh melalui penggunaan software khusus penginderaan jauh yang diterapkan padakomputer.

f. Pengguna Data

Pengguna data merupakan komponen akhir yang penting dalam sistem inderaja, yaitu orang atau lembaga yang memanfaatkan hasil inderaja. Jika tidak ada pengguna, maka data inderaja tidak ada manfaatnya. Salah satu lembaga yang menggunakan data inderaja misalnya adalah:

- Bidang militer
- Bidang kependudukan
- Bidang pemetaan
- Bidang meteorologi dan klimatologi

7.5 Keunggulan, Keterbatasan dan Kelemahan Penginderaan Jauh

a. Keunggulan Inderaja

Menurut Sutanto (1994:18-23), penggunaan penginderaan jauh baik diukur dari jumlah bidang penggunaannya maupun dari frekuensi penggunaannya pada tiap bidang mengalami pengingkatan dengan pesat. Hal ini disebabkan oleh beberapa faktor antara lain :

- Citra menggambarkan obyek, daerah, dan gejala di permukaan bumi dengan; wujud dan letak obyek yang mirip ujud dan letak di permukaan bumi, relatif lengkap, meliputi daerah yang luas, serta bersifat permanen.

- Dari jenis citra tertentu dapat ditimbulkan gambaran tiga dimensional apabila pengamatannya dilakukan dengan alat yang disebut stereoskop.
- Karaktersitik obyek yang tidak tampak dapat diwujudkan dalam bentukcitra sehingga dimungkinkan pengenalan obyeknya.
- Citra dapat dibuat secara cepat meskipun untuk daerah yang sulit dijelajahi secara terestrial.
- Merupakan satu-satunya cara untuk pemetaan daerah bencana.
- Citra sering dibuat dengan periode ulang yang pendek.

b. Keterbatasan Inderaja

Berupa ketersediaan citra SLAR yang belum sebanyak ketersediaan citra lainnya. Dari citra yang ada juga belum banyak diketahui serta dimanfaatkan (Lillesand dan Kiefer, 1979). Di samping itu jugaharganya yang relative mahal dari pengadaan citra lainnya (Curran, 1985).

c. Kelemahan Inderaja

Walaupun mempunyai banyak kelebihan, penginderaan jauh juga memiliki kelemahan antara lain sebagai berikut

- Orang yang menggunakan harus memiliki keahlian khusus;
- Peralatan yang digunakan mahal;
- Sulit untuk memperoleh citra foto ataupun citra nonfoto.

A. Manfaat Penginderaan Jauh

a. Bidang Kelautan (Seasat, MOS)

- Pengamatan sifat fisis air laut.
- Pengamatan pasang surut air laut dan gelombang laut.

b. Pemetaan perubahan pantai, abrasi, sedimentasi, dan lain-lain

- Bidang hidrologi (Landsat, SPOT)
- Pemanfaatan daerah aliran sungai (DAS) dan konservasi sungai.
- Pemetaan sungai dan studi sedimentasi sungai.
- Pemanfaatan luas daerah dan intensitas banjir.

c. Bidang geologi

- Menentukan struktur geologi dan macamnya.
- Pemantauan daerah bencana (gempa, kebakaran) dan pemantauan debu vulkanik.
- Pemantauan distribusi sumber daya alam.
- Pemantauan pencemaran laut dan lapisan minyak di laut.
- Pemanfaatan di bidang pertahanan dan militer.
- Pemantauan permukaan, di samping pemotretan dengan pesawat terbang dan aplikasisistem informasi geografi (SIG).

d. Bidang meteorologi dan klimatologi (NOAA)

- Membantu analisis cuaca dengan menentukan daerah tekanan rendah dan daerah bertekanan tinggi, daerah hujan, dan badai siklon.
- Mengetahui sistem atau pola angin permukaan.

- Permodelan meteorologi dan data klimatologi.
- Untuk pengamatan iklim suatu daerah melalui pengamatan tingkat kewarnaan dan kandungan air di udara.

e. Bidang oseanografi

- Pengamatan sifat fisis air seperti suhu, warna, kadar garam dan arus laut.
- Pengamatan pasang srut dengan gelombang laut (tinggi, frekuensi, arah).
- Mencari distribusi suhu permukaan.
- Studi perubahan pasir pantai akibat erosi dan sedimentasi

Bab 8

Memahami Visualisasi Data

8.1 Pengantar

Visualisasi data atau *data visualization* berhubungan dengan komunikasi data. Visualisasi data adalah teknik mengkomunikasikan data atau informasi dengan membuatnya ke dalam objek visual ke dalam grafik. Contohnya seperti titik, batang, garis, dan lain-lain. Dalam era digital, kehadiran data sangat lah penting. Data yang tersebar sangatlah banyak. Agar bisa terserap menjadi informasi yang berguna dan jelas, maka butuh penyajian data.

Sebuah penelitian menunjukkan, 80% seseorang memahami dari apa yang dilihat dalam visual, sedangkan 20% melalui yang dibaca. Penelitian lain juga menunjukkan, konten visual mampu diproses manusia sebanyak 60 ribu kali lebih cepat dari konten lain (Aprillia, 2022).

Keberadaan visualisasi data bisa dikatakan sama dengan dengan komunikasi. Jika komunikasi data yang dilakukan baik, maka penyajian data dan penyampaian informasi akan baik pula. Tetapi jika sebaliknya, maka hal itu pun juga bisa menjadi tidak baik. Visualisasi yang baik bisa berfokus memberikan jawaban yang jelas dan tidak terlalu detail (Muharni & Candra, 2022).

Visualisasi data bisa menggunakan dashboard yang mana teks, korelasi, dan pola yang tidak terdeteksi dapat divisualisasikan dengan menggunakan perangkat lunak. Visualisasi tidak hanya mengubah data menjadi grafik visual, tetapi juga perlu rencana. Di mana, tiap jenis data butuh visualisasi sesuai kebutuhan.

8.2 Tujuan, Fungsi, dan Jenis

Kehadiran visualisasi mempermudah peneliti untuk melihat data sehingga bisa mengamati simulasi dan komputasi, juga memperkaya proses penemuan ilmiah dan mengembangkan pemahaman lebih mendalam. Inti dari visualisasi sebetulnya untuk menemukan metode terbaik dan menampilkan data. Maka, tujuan dari visualisasi data adalah:

1. Mengeksplor

   Dalam visualisasi, eksplorasi dilakukan terhadap data atau informasi yang bisa digunakan sebagai salah satu bagian dari elemen pengambilan keputusan.

2. Menghitung

   Dalam visualisasi, menghitung diartikan sebagai kegiatan analisa data dalam bentuk grafik atau table. Dengan begitu, manajemen hanya perlu melakukan pengembilan keputusan dari data yang sudah terhitung.

3. Menyampaikan

   Dalam visualisasi, data mentah yang diolah dalam bentuk grafik adalah bentuk penyampaian dengan cara pendekatan visual yang dapat membuat orang melihat gambar dengan lebih mudah. Sebab, data dalam bentuk grafik terkesan lebih mudah dipahami.

Dalam penelitian lain, visualisasi data memiliki tujuan dua hal: (1) Analisa; bagaimana mengerti data, mengambil informasi dan bersifat komprehensif; (2) komunikasi; bagaimana mengkomunikasikan informasi, melibatkan

penyederhanaan dan nilai rasa (Ilyas & Pudjiantoro, 2015). Dengan tujuan dari visualisasi data, maka visualisasi data menjadi penting sebagai komunikasi data.

Biasanya, visualisasi data sangat berkaitan dengan dunia bisnis. Data ditampilkan melalui visualisasi supaya mudah dipahami pengguna, sekaligus juga memberikan nilai makna. Bagi para pebisnis, hal itu menjadi penting lantaran bisa membantu pebisnis mengambil keputusan. Di samping itu, visualisasi data juga menjadi tren yang baik dalam menyampaikan informasi disbanding hanya menggunakan teks atau tabel.

Hasil dari visualisasi data bisa mennjadi suatu *novelty* atau kebaruan bagi karya pekerjaan. Hal ini juga bisa memperkuat daya pikir manusia untuk berkreasi melalui data. Sebab, dengan membuat visualisasi data, ini bisa menjadi alat komunikasi yang efektif. Karena terintegrasi teknologi, maka pengembangan visualisasi data berkembang lebih pesat dengan paduan desain visual yang terus berkembang.

Selain tujuan, visualisasi data memiliki beberapa fungsi yang juga penting untuk diketahui. Beberapa fungsi terlampir sebagai berikut:

1. Permudah pemahaman data

   Para audiens lebih mudah memahami visualisasi data yang ditampilkan sehingga bisa mengambil keputusan yang lebih akurat. Sebab, visualisasi data menyorot informasi penting.

2. Meningkatkan pemahaman tentang operasional bisnis

   Bisnis perlu kumpulan data yang tidak sedikit. Visualisasi data dapat meningkatkan pemahaman tentang opersional bisnis, utamanya dari segi strategi pemasaran dan minat produk yang bisa dilakukan ke depan.

3. Meningkatkan nilai produk atau jasa

   Sebagian pengguna tidak punya waktu atau kesabaran untuk mempelajari alat. Kehadiran visualisasi data dibutuhkan untuk memberi pengguna kenyamanan dan intuitif.

4. Memberdayakan orang dengan advanced analytics

   Visualisasi data dapat membentuk bisnis atau organisasi dapat mengembangkan wawasan lebih mendalam. Selain itu, bisa menemukan pola tersembunyi, dan bertindak berdasar peluang bisnis dengan nilai tinggi.

Visualisasi data bisa menggambarkan relasi dua pola antara variabel. Dengan adanya visualisasi data, maka para penggunanya bisa melihat koneksi antara data yang bersifat multidimensi. Sebab bagaimanapun, visualisasi data itu mengkomunikasikan informasi secara jelas dan efektif melalui sarana grafis. Visualisasi yang baik membutuhkan proses visualisasi data yang meliputi *acquire, parse, filter, mine, represent, refine*, dan *interact* (Madyatmadja, et al., 2021).

Acquire adalah proses pengumpulan data dari beragam sumber dari file penyimpan atau jaringan. Tahapan ini fokus bagaimana agar data diperoleh. Parse adalah proses penyesuaian data ke format yang ditentukan akan dikategorikan ke dalam beberapa kategori agar dapat dibaca dan dibedakan dengan lainnya. Filter merupakan proses seleksi data dengan cara menghapus data yang tidak perlu.

Mine merupakan proses penerapan metode disiplin ilmu statistika dan data mining. Proses ini sebagai langkah mencari pola atau dijabarkan pada konteks matematis. Represent merupakan proses mengubah data dalam bentuk visual. Tahap ini menunjukkan bentuk dasar yang diambil. Refine merupakan proses meningkatkan hasil representasi agar terlihat menarik. Interact merupakan proses penambahan metode untuk

manipulasi data atau mengendalikan fitur yang terlihat dengan kata lain bisa ditampilkan sesuai kehendak pengguna.

Dalam hal ini, visualisasi data juga memiliki beragam jenis. Jenis ini menunjukkan tipe-tipe visualisasi data yang cocok digunakan berdasarkan kebutuhan. Jenis visualisasi data antara lain:

1. Visualisasi data sementara (temporal data visualization)

   Visualisasi data sementara/temporal ditujukan sebagai hasil dari rangkaian data yang berbentuk linear atau satu dimensi saja. Ciri utama visualisasi data sementara berupa garis yang bermula dan berakhir di titik tertentu. Contohnya seperti line chart, bar chart, polar area diagrams, scatter plots time series sequences, dan timelines.

2. Visualisasi data hierarkis (hierarchical data visualizations)

   Visualisasi ini digunakan untuk menunjukkan hubungan antarkelompok terhadap kelompok lain yang lebih besar.jenis visualisasi ini cocok untuk memunculkan data-data baru yang berasal dari suatu penyebab. Contohnya, diagram pohon dan grafik batang.

3. Visualisasi data jaringan (network data visualizations)

   Visualisasi ini lebih menekankan pada hubungan antara entitas dan tautan. Jenis ini adalah sekumpulan data yang saling berpengaruh satu sama lain. Contohnya seperti alluvial diagrams, word cloud, node-link diagrams, hingga matrix chart.

4. Visualisasi data multidimensi (multidimensional data visualizations)

   Visualisai data ini lebih menekankan variable yang banyak. Keberadaan visualisasi data multidimensi biasanya memiliki tampilan lebih menarik. Contoh visualisasi ini antara lain histograms, stacked bar, dan pie charts.

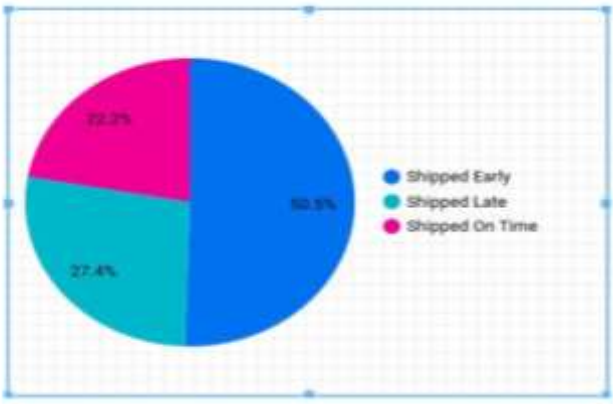
5. Visualisasi data geospasial (geospatial data visualizations)

   Biasanya, visualisasi ini memberikan wujud nyata dari benda atau ruang yang punya data untuk ditampilkan. Contohnya, cartogram, heat map, flow map, hingga density map.

Contoh

Sebagai percontohan, kita bisa melihat bagaimana penggunaan pie chart dalam google data studio. Pie chart atau diagram lingkaran adalah grafik statistic lingkaran yang dibagi menjadi beberapa irisan dan luasnya tergantung pada komposisi antar kategori dari data yang dimiliki. Ciri utamanya adalah bisa memperlihatkan perbandingan ukuran data besar sector secara langsung.

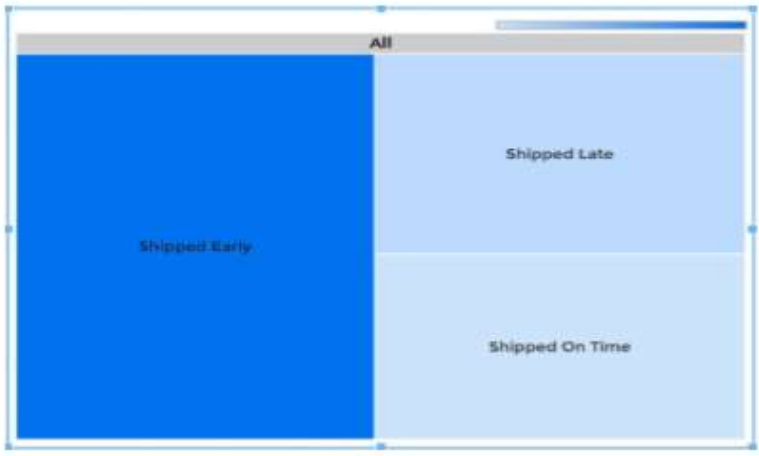
Berikut percontohan pembuatannya. Dalam google data studio, buka page 1 Bar and scorecard. Masukkan pie chart pada dashboard. Lalu, letakkan di bawah column chart. Pada sidebar, masukkan ship status pada dimension.ubah metric jadi sales. Dari sini, bisa dilihat berapa banyak barang yang dikirim tepat waktu, lebih awal, ataupun telat.



Gambar 8.1 contoh pie chart

Sumber: (Muharni & Candra, 2022)

Pada bentuk yang lain, penulis juga memberikan contoh dari treemap. Treemap adalah cara alternative memvisualisasikan struktur hierarki diagram tree, sekaligus juga menampilkan jumlah tiap kategori melalui ukuran area. Tiap kategori berisi area persegi Panjang dengan sub kategori di dalamnya. Untuk prosesnya dalam google data studio, buat dashboard baru dan beri nama treemap, lalu buat chart baru dengan tipe treemap.



Gambar 8.2 contoh treemap

Sumber: (Muharni & Candra, 2022)

Tambahkan category dan ship status ke dalam dimension. Tambahkan pula record count ke dalam metric. Hasilnya akan jadi seperti ini.

Gambar 8.2 contoh treemap

Sumber: (Muharni & Candra, 2022)

Visualisasi data di sini, salah satunya, menggunakan google data studio. Ini menjadi salah satu solusi alternative dalam mengoperasionalisasikan visualisasi data. Dalam Google data studio, terdapat berbagai sumber daya sehingga memudahkan pengguna untuk mengintegrasikan laporan dari berbagai sumber yang ada. Sehingga, pengguna bisa menangkap rangkuman data yang diperoleh melalui visualisasi.

8.3 Penutup

Visualisasi data merupakan alat komunikasi data yang efektif. Sejauh ini, riset membuktikan, dijelaskan sebelumnya, bahwa manusia lebih mudah memahami visualisasi data disbanding kumpulan teks atau sekadar kumpulan angka. Visualisasi data mampu memberikan rekaman ke otak manusia secara lebih baik.

Tujuan utama dari visualisasi data itu lebih kepada sebagai alat komunikasi data yang efektif untuk manusia. Fungsinya pun bermacam-macam: mempermudah pemahaman data; meningkatkan pemahaman operasional bisnis; meningkatkan nilai produk atau jasa; memberdayakan dengan

advanced analytics. Begitu juga jenis-jenisnya: visualisasi data sementara, jaringan, hierarkis, multidimensional, dan geospasial.

Ada pun contoh dari visualisasi data sebetulnya ada beragam bentuk. Hanya saja, dalam tulisan ini, cuma beberapa yang ditampilkan sebagai bagian dari percontohan. Pada umumnya, melalui contoh yang ditampilkan, visualisasi data condong menggunakan warna-warni. Hal ini ditujukan agar lebih memberikan ketertarikan di indra mata. Ketertarikan itu bisa memberikan pemahaman lebih terhadap data yang dipaparkan.

Bab 9

Quantitative Mini Research: Analisis Regresi Berorientasi Kasus Kesehatan, Ketimpangan, Dan Kemiskinan

9.1 Latar Belakang

Penelitian makro-komparatif adalah dunia mikro sosiologis yang mencerminkan beberapa kontradiksi dan ketegangan itu menjiwai sosiologi sebagai suatu disiplin. Para peneliti yang berorientasi pada kasus dan berorientasi variabel terlibat dalam perdebatan yang sudah berlangsung lama berulang kali menjadi terkenal (Clausen dan Mjoset 2007; Goertz dan Mahoney 2012; Goldthorpe 1997; Kenworthy 2007; King, Keohane, dan Verba 1994; Ragin 1987, 2000, 2008; Rubinson 2019; Shalev 2007; Verba 1967). Dengan tingkat perkiraan, seseorang dapat meringkas posisi sebagai berikut. Peneliti berorientasi kasus mengklaim mengejar pendekatan khusus untuk ilmu sosial yang diatur oleh seperangkat asumsi tertentu (Ragin 1997).

Peneliti berorientasi variabel, sebaliknya, sarankan agar analisis berorientasi kasus dapat ditingkatkan dengan mengikuti standar kuantitatif secara lebih ketat (King et al. 1994). Para pendukung persuasi yang terakhir bisa dibilang memimpin perlombaan, seperti yang ditunjukkan oleh dominasi studi kuantitatif yang diterbitkan dalam jurnal

sosiologis. Namun, penulis berpendapat bahwa dalam mengenali logika yang mendasarinya yang berlaku untuk analisis kualitatif dan kuantitatif, yaitu adalah, logika inferensi (King et al. 1994:3), mungkin banyak gagal untuk mengenali logika dasar umum kedua, yaitu, logika kontribusi kasus.

Sebuah paradoks metodologi mencirikan penelitian makro-komparatif: hal itu secara rutin melanggar asumsi yang mendasari metode dominannya, analisis regresi berganda. Peneliti komparatif memiliki minat substantif dalam kasus, tetapi sebagian besar kasus tidak terlihat dalam analisis regresi. Peneliti jarang mengenali ketidakcocokan antara tujuan penelitian makro-komparatif dan tuntutan metode regresi dan kadang-kadang akhirnya menarik dalam sengketa berat atas efek variabel tertentu.

Contoh yang baik adalah hubungan kontroversial antara ketimpangan pendapatan dan kesehatan. Di sini, penulis menawarkan metode inovatif yang menggabungkan orientasi variabel dan pendekatan berorientasi kasus dengan memutar model regresi kuadrat terkecil biasa "luar dalam". Para penulis memperkirakan kontribusi khusus kasus terhadap estimasi koefisien regresi. Mereka menganalisis kembali data tentang ketimpangan pendapatan, kemiskinan, dan harapan hidup di 20 negara kaya. Beberapa spesifikasi model bergantung terutama pada dua negara dengan nilai-nilai pada hasil yang sangat besar dan tidak konsisten dengan ekspektasi teoretis konvensional.

9.2 Analisis Makro-Komparatif dan Analisis Regresi Berganda

Analisis makro-komparatif kebijakan sosial, ekonomi politik, dan rezim kesejahteraan secara rutin menggunakan metodologi umum, yakni analisis regresi berganda yang asumsinya adalah terus-menerus dilanggar dalam perbandingan n kecil (Shalev 2007). Selain keterbatasan statistik, orang dapat berargumen bahwa masalah yang utama di sini adalah bahwa analisis regresi berganda cenderung membuat

terlihat persis apa yang merupakan fokus komparatif investigasi, yaitu kasus-kasus. Para sarjana telah mengusulkan sejumlah alternatif, seperti analisis lintas-tabular berteknologi rendah, analisis faktor, analisis kluster (Shalev 2007), dan analisis komparatif kualitatif, baik dalam himpunan biner maupun versi himpunan kusut (Ragin 1987, 2000, 2008). Meskipun semua teknik ini memiliki keunggulan, mereka semua, pada tingkat yang berbeda-beda, gagal untuk menjadi benar-benar arus utama, setidaknya dalam sosiologi Amerika yang mengadopsi analisis statistik sebagai penanda kekakuan ilmiah (Camic dan Xie 1994; Leahey 2005; Liao 2014).

Meskipun beberapa analisis komparatif menganjurkan untuk tidak menggunakan teknik regresi sama sekali, yang lain percaya masalahnya bukan pada metode itu sendiri tetapi penerapannya di fakta, "analisis statistik banyak digunakan secara kejam, jika tidak kebanyakan ilmu sosial komparatif" (Scruggs 2007:310). Kenworthy (2007) mengusulkan rencana untuk meningkatkan penggunaan regresi berganda dalam studi banding yang melibatkan presentasi grafis yang lebih transparan dari hasil, dan lebih memperhatikan arah, besaran, dan kekokohan koefisien. Strategi-strategi ini merupakan peningkatan, tetapi mungkin masih gagal membuat kasus terlihat. Bahkan, Kenworthy mencatat, "dalam artikel komparatif makro kuantitatif prototipe, regresi adalah titik awal dan akhir dari analisis. Saya ingin melihat lebih banyak makalah di mana regresi digunakan untuk menginformasikan diskusi kasus".

Breiger dan rekan (Breiger et al. 2011; Breiger dan Melamed 2014; Melamed, Breiger, dan Schoon 2012; Melamed, Schoon, dkk. 2012) mengambil tantangan ini dan mengembangkan metodologi yang memungkinkan peneliti untuk mengungkap berbagai cerita berbaring di bawah analisis regresi. Faktanya, peneliti pertama-tama dapat memperkirakan model regresi tradisional dan kemudian mengubahnya "dalam

ke luar" untuk menunjukkan kontribusi kasus-spesifik untuk estimasi koefisien. Metode ini secara faktual menjembatani metode kuantitatif dan kualitatif dan dalam pengertian ini mengatasi narasi hegemonik analisis regresi. Penulis menerapkan pendekatan ini pada studi tentang ketimpangan pendapatan, kemiskinan, dan kesehatan yang literaturnya penulis ulas secara singkat.

9.3 Ketimpangan dan Kesehatan

Pada 1990-an serangkaian artikel diterbitkan di Jurnal Medis Inggris (Kaplan dkk. 1996; Kennedy dkk. 1998; Kennedy, Kawachi, dan Prothrow-Stith 1996; Wilkinson 1992), mengikuti studi perintis di akhir 1970-an (Rodgers 1979), menunjukkan bahwa tingkat agregat ketimpangan pendapatan berkorelasi dengan langkah-langkah kesehatan penduduk, baik di seluruh negara makmur dan di seluruh negara bagian A.S., bahkan setelah mengendalikan sejumlah faktor, seperti pendapatan rata-rata dan tingkat kemiskinan. Argumen bahwa ketidaksetaraan membuat kita sakit berada di bawah kulit kita sangat kuat dan itu beresonansi dengan gagasan dan bukti yang dihasilkan dalam "kumpulan ilmu sosial yang sangat besar dan berkembang dengan baik" (Beckfield dan Krieger 2009:153) yang ditandai dengan banyaknya publikasi yang melaporkan konsekuensi negatif dari ketidaksetaraan (Jencks 2002).

Singkatnya, hipotesis ketimpangan pendapatan atau *income inequality hypothesis* (IIH) menyatakan bahwa di dunia yang makmur, ketimpangan pendapatan merugikan kesehatan bersih populasi dari determinan sosial kesehatan lainnya. Ratusan penelitian telah menguji IIH dan menggunakan berbagai macam metode (Leigh et al. 2009; Mullahy, Robert, dan Wolfe 2008). Perbandingan ekologi di tingkat nasional dan subnasional tingkat sangat populer karena IIH menyarankan ketimpangan pendapatan rata-rata dikaitkan dengan rata-rata kesehatan penduduk. Padahal, buku yang memindahkan IIH

dari debat akademik eksklusif ke publik yang lebih besar percakapan politik, *The Spirit Level* (Wilkinson dan Pickett 2009), juga menyajikan perbandingan ekologis.

Setelah antusiasme awal, IIH memperoleh cukup banyak kritikus (Avendano dan Hessel 2015; Beckfield 2004; Goldthorpe 2010; Lynch dkk. 2004). Satu pertanyaan adalah apakah ketimpangan pendapatan sebenarnya terkait dengan kesehatan. Bahkan, ulasan memperkirakan bahwa 30 persen dari bukti yang diterbitkan tidak mendukung IIH (Pickett dan Wilkinson 2015; Wilkinson dan Pickett 2006). Masalah lainnya adalah bagaimana menafsirkan asosiasi. Dua narasi muncul (Lynch et al. 2000). Yang pertama mengusulkan sebuah hubungan interpretasi psikologis: ketidaksetaraan meningkatkan kecemasan status yang berubah menjadi stres kronis yang menyebabkan kesehatan yang buruk. Yang kedua menganggap ketimpangan pendapatan sebagai bagian dari sekelompok kondisi neomaterial (misalnya, kemiskinan, investasi publik, dan kesejahteraan) yang semuanya berkontribusi dalam membentuk populasi kesehatan.

Satu studi baru-baru ini mencoba pendekatan yang berbeda: alih-alih memainkan ketidaksetaraan dan kemiskinan satu sama lain, Rambotti (2015) menyelidiki apakah mereka berinteraksi. Di sini, penulis mereproduksi analisis ini dan kemudian membalikkannya, memeriksa kontribusi setiap kasus (masyarakat nasional) terhadap koefisien regresi, dan ini membawa penulis pada temuan penting yang tidak diantisipasi dalam penelitian sebelumnya.

9.4 Analisis Regresi OLS

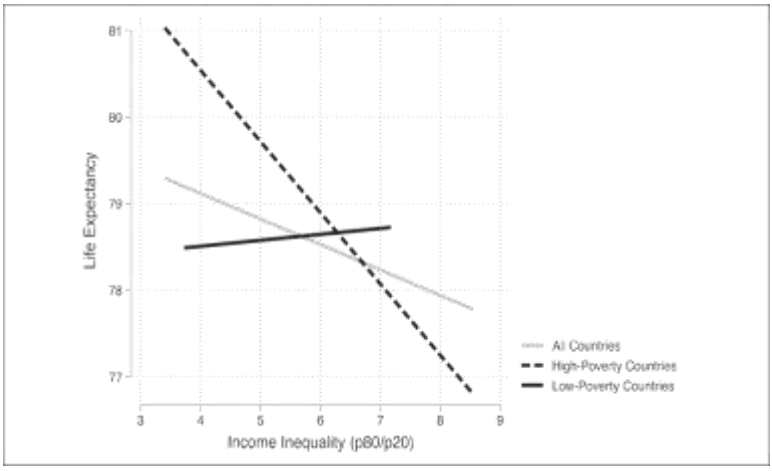
Temuan ini mereproduksi penelitian yang diterbitkan sebelumnya (Rambotti 2015) dengan satu perbedaan: di sini penulis menggunakan variabel standar dan (dengan demikian) koefisien regresi standar. Ini akan menyederhanakan interpretasi kontribusi spesifik kasus terhadap koefisien itu

sendiri. Penulis mengabaikan intersep yang dalam model standar sama dengan nol. Hasilnya identik dengan yang ada di artikel asli dan interpretasi tipikal mereka akan menekankan signifikansi statistik (atau kekurangannya) dari setiap koefisien regresi: tidak ada ketimpangan pendapatan baik (model 1) maupun kemiskinan (model 2) signifikan terkait dengan harapan hidup; ketika kedua variabel tersebut dipertimbangkan bersama-sama, tetapi ketimpangan menjadi signifikan dalam regresi berganda (model 3). Selain itu, kapan istilah interaksi antara dua variabel ditambahkan ke model (model 4), interaksi signifikan, dan kekuatan penjelas model meningkat secara substansial (disesuaikan R^2 = 39 persen).

Tabel 9.1: Koefisien Standar dan Kesalahan Standar (dalam Tanda kurung) dari Model Harapan Hidup Ordinary Least Square. *$p < 0.05$ dan **$p < 0.01$ (tes dua sisi).

	Model 1	**Model 2**	**Model 3**	**Model 4**
Ketimpangan pendapatan	-0.36		-0.73*	-0.68*
	(0.21)		(0.27)	(0.24)
Kemiskinan		0.07	0.56	0.75**
		(0.23)	(0.27)	(0.25)
Ketimpangan × Kemiskinan				-0.49*
				(0.20)
N	20	20	20	20
R^2	0.13	0.00	0.22	0.39
F	2.75	0.08	3.80	5.22

Interpretasi substantif dari hasil ini, terlihat pada Gambar 1, adalah bahwa ketimpangan pendapatan yang lebih tinggi dikaitkan dengan kehidupan yang harapan hidupnya lebih rendah hanya di negara-negara dengan tingkat kemiskinan tinggi (yaitu, negara-negara yang standar ukuran kemiskinan di atas nol). Sebaliknya, ketimpangan pendapatan sama sekali tidak terkait dengan harapan hidup di negara-negara miskin rendah (12 negara dalam sampel penulis yang standar ukuran kemiskinannya di bawah nol).



Gambar 9.1 : Harapan hidup berdasarkan ketimpangan pendapatan (semua negara, kemiskinan rendah, dan kemiskinan tinggi). Diadaptasi dari Rambotti (2015).

Sejumlah tes dilakukan untuk memeriksa model ini agar tidak melanggar asumsi penting dan untuk memeriksa bahwa analisisnya tidak sensitif terhadap *outlier* ekstrim. Misalnya, karena korelasi antara ketimpangan pendapatan dan kemiskinan, penting untuk mengevaluasi kolinearitas dalam model 3. Faktor inflasi varian jauh di bawah ambang batas 10. Tes visual dan formal menunjukkan bahwa model lengkap menyajikan residu yang terdistribusi secara normal dan memiliki varian konstan. *Resampling* Jackknife menunjukkan bahwa kesalahan standar tidak terlalu bias oleh kemungkinan

adanya *outlier* yang parah (Quenouille 1949, 1956; Tukey 1958). Semua hal dipertimbangkan, modelnya berfungsi baik pada tes ini. Dikatakan demikian, karakteristik analisis ini (analisis kuantitatif dari sekumpulan kecil pengamatan non-acak) memerlukan penyelidikan lebih lanjut dari kontribusi khusus kasus yang merupakan tujuan dari penelitian ini.

9.5 Membalikkan Regresi Luar Dalam

Sesuai dengan model 4 pada Tabel 2, matriks X adalah matriks kasus-per-variabel 20 × 3. Menggunakan persamaan 2, penulis memperoleh koefisien β yang sama terlihat pada model 4 OLS di atas: −0.68, 0.75, dan −0.49. Lebih menarik lagi, jika kita mengganti y dengan Y = diag(y) (lihat persamaan 3), kita dekomposisikan koefisien regresi menjadi kontribusi spesifik kasus yang unik, satu per negara, seperti yang ditunjukkan pada Tabel 9.2.

Tabel 9.2. Koefisien *Ordinary Least Square* sebagai Penjumlahan Kasus (Model 4)

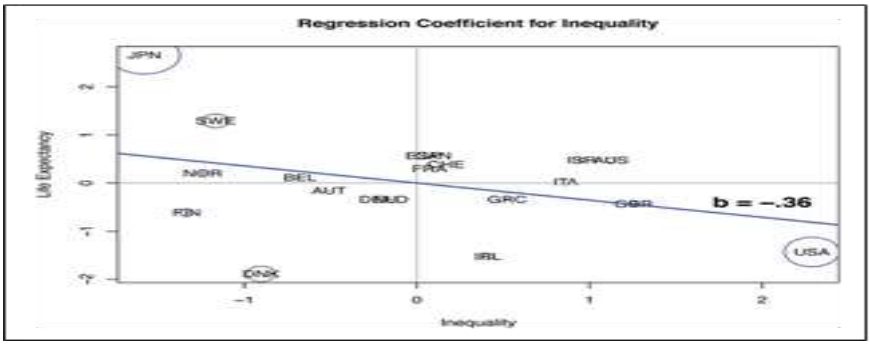
Negara	Ketimpangan Pendapatan	Kemiskinan	Ketimpangan × Kemiskinan
AUS	0.06	-0.03	-0.02
AUT	0.00	0.01	0.00
BEL	0.00	0.00	0.00
CAN	-0.02	0.05	-0.03
CHE	0.02	-0.02	-0.01
DEU	-0.01	0.01	0.00
DNK	0.04	0.15	-0.12

ESP	-0.02	0.06	-0.03
FIN	0.04	0.03	-0.05
FRA	0.01	-0.01	-0.01
GBR	-0.06	0.03	0.02
GRC	-0.01	-0.01	0.01
IRL	-0.05	-0.01	0.06
ISR	-0.03	0.08	0.01
ITA	0.00	0.00	0.00
JPN	-0.45	0.45	-0.19
NLD	-0.01	0.02	0.00
NOR	-0.02	0.00	0.01
SWE	-0.07	-0.07	0.09
USA	-0.09	0.01	-0.24
Total	**-0.68**	**0.75**	**-0.49**

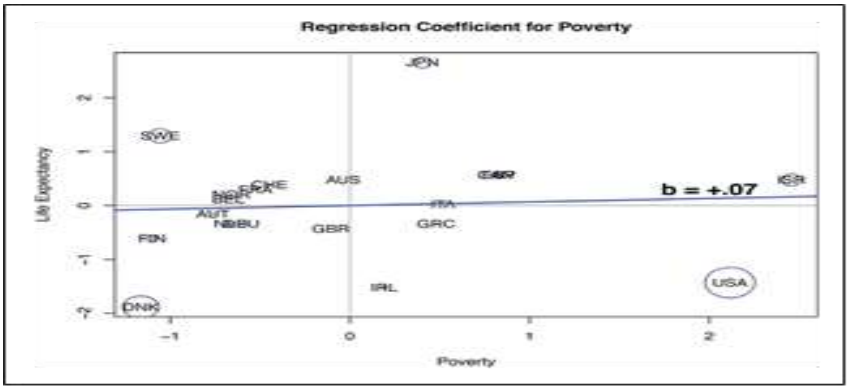
Tabel di atas menunjukkan cara yang agak baru untuk memvisualisasikan analisis regresi yang mengungkap berbagai cerita yang ada di bawah analisis regresi. Sangat sedikit preseden di literatur menguraikan koefisien regresi oleh kasus kelompok dalam analisis kesejahteraan sosial dan jaringan *homophily* (Melamed, Breiger, et al. 2012), dan organisasi ekstremis kekerasan (Breiger dan Melamed 2014).

Penulis meminta perhatian pembaca pada beberapa fitur Tabel dibawah. Pertama, perhatikan bahwa angka terakhir di setiap kolom adalah hasil penjumlahan dari 20 nomor (satu untuk setiap negara) di atasnya yang mana menunjukkan bagaimana koefisien regresi adalah penjumlahan lintas kontribusi kasus spesifik. Faktanya, −0.68, 0.75, dan −0.49 adalah koefisien standar OLS untuk efek dari masing-masing, ketimpangan pendapatan, kemiskinan, dan interaksi ketimpangan dan kemiskinan terhadap harapan hidup (lihat model 4 pada Tabel 2 untuk perbandingan). Kedua, jika kita fokus pada kolom terakhir, yaitu menunjukkan dekomposisi koefisien interaksi, kita dapat mengamati bahwa interaksi ini sebagian besar disebabkan oleh kontribusi segelintir kasus. Secara khusus, Amerika Serikat, Jepang, dan Denmark berkontribusi negatif (masing-masing −0.24, −0.19, dan −0.12), sedangkan Swedia menjadi kontribusi positif yang terbesar (0,09). Jepang dan Denmark menawarkan kontribusi tertinggi terhadap efek utama kemiskinan (masing-masing, 0,45 dan 0,15). Akhirnya, Jepang memberikan kontribusi paling besar untuk efek utama dari ketimpangan (−0.45). Penulis mengusulkan untuk memberi label kontribusi unik ini sebagai *intensitas*.

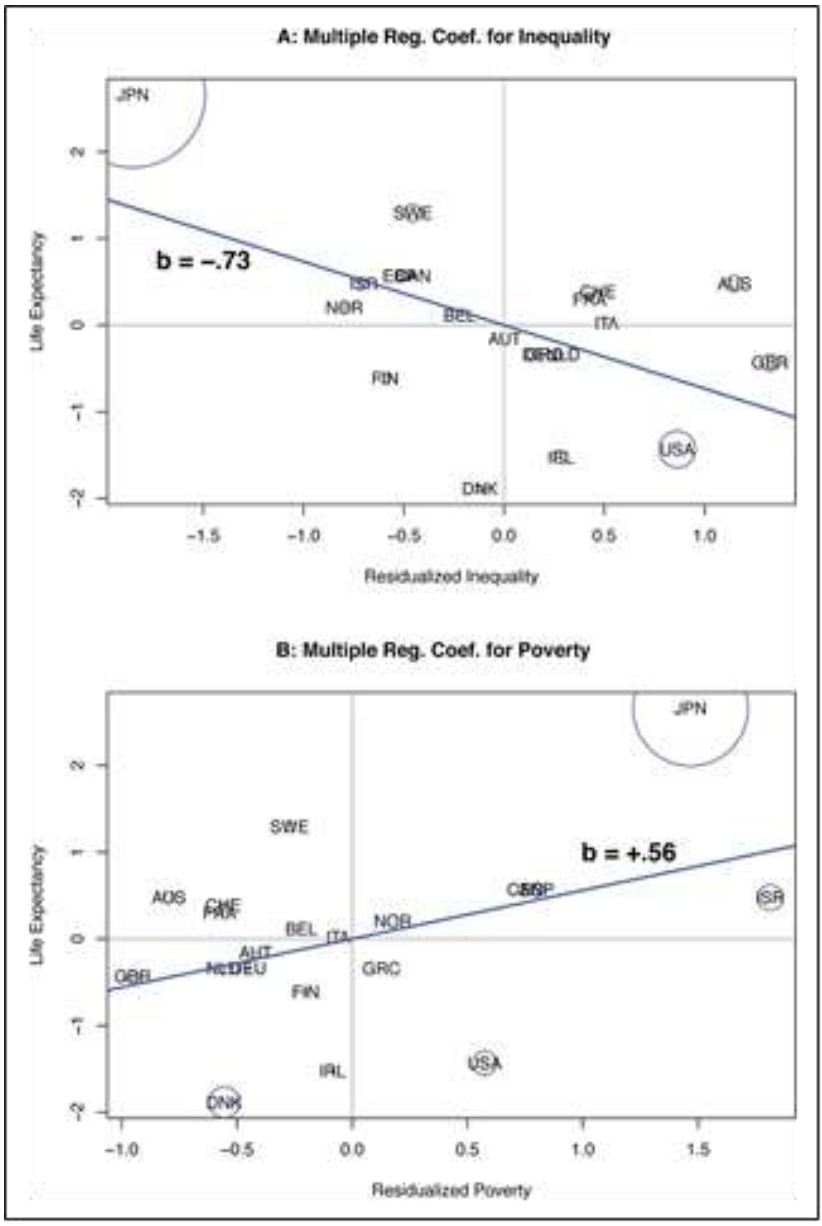
Dengan demikian, 4 dari 20 negara mendominasi hubungan Model OLS secara penuh yang menunjukkan jumlah penjelasan variasi terbesar (r^2 = 0.39 untuk model 4 pada Tabel 2). Apa yang terjadi ketika kita melihat model lain? Dan bagaimana kita menafsirkan kontribusi positif dan negatif terhadap koefisien regresi? Untuk menjawab pertanyaan ini, penulis menguraikan koefisien standar dalam model 1 sampai 3. Namun, untuk memungkinkan pemahaman data yang lebih cepat dan intuitif, penulis menyajikan intensitas spesifik negara ini dalam bentuk visual. Setiap dari grafik berikut (Gambar 2–5) mewakili sebar dari harapan hidup negara versus variabel prediktor.



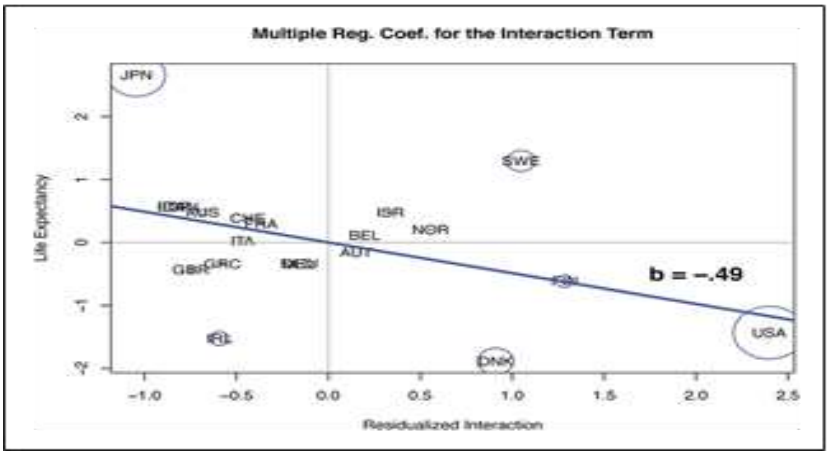
Gambar 9.2. Dekomposisi koefisien, model 1: ketimpangan (−0.36). Catatan: Ukuran lingkaran sebanding dengan kontribusi terhadap koefisien regresi.



Gambar 9.3. Dekomposisi koefisien, model 2: kemiskinan (+0.07). Catatan: Ukuran lingkaran sebanding dengan kontribusi terhadap koefisien regresi.



Gambar 9.4. Dekomposisi koefisien, model 3: ketimpangan (−0.73) (A) dan kemiskinan (+0.56) (B). Catatan: Ukuran lingkaran sebanding dengan kontribusi ke masing-masing koefisien regresi berganda.



Gambar 9.5. Dekomposisi koefisien, model 4: interaksi suku (−0.49). Catatan: Ukuran lingkaran sebanding dengan kontribusi ke kelipatan koefisien regresi untuk istilah interaksi.

Gambar diatas, misalnya, memplot harapan hidup ketimpangan (kedua variabel dalam bentuk standar atau z-score dengan rata-rata 0 dan standar deviasi 1). Koefisien regresi keseluruhan adalah −0.36. Ukuran lingkaran pada Gambar 2 adalah sebanding dengan kontribusi negara terhadap koefisien regresi keseluruhan. Kontribusi didefinisikan dalam diskusi penulis tentang persamaan 3 dan angka-angka ini diberikan untuk dipilih negara pada Tabel 4. Dengan menggunakan Gambar 2, kita dapat mengilustrasikan apa yang dimaksud dengan intensitas. Pertimbangkan, misalnya, Swedia (SWE) pada Gambar 2. Swedia memiliki ketimpangan yang lebih kecil daripada rata-rata negara yang penulis pelajari (skornya pada sumbu x Gambar 2 adalah −1.1657). Harapan hidup Swedia di atas rata-rata (skornya pada sumbu y adalah +1,2952). Sekarang pertimbangkan persegi panjang yang dibentuk dengan mengambil koordinat Swedia (x = −1.1657, y = +1.2952) dan koordinat asal (x = 0, y = 0) sebagai sudut yang berlawanan. Daerah persegi panjang ini adalah perkalian dari −1,1657 dan +1,2952, yaitu −1.5099 (penulis lampirkan tanda produk ke area

persegi panjang) yang merupakan konstanta (n − 1 = 19) dikalikan dengan berapa penulis menyebut "kontribusi" Swedia pada koefisien regresi.

Kontribusi ini adalah −1.5099/19 = −.0795, yaitu dilaporkan ke dua tempat desimal pada Tabel. Menurut definisi dari koefisien regresi, jumlah kontribusi sama dengan koefisien regresi. Kontribusi tersebut dengan tanda positif dihasilkan oleh negara-negara yang nilai x dan y-nya berada di sisi yang sama dengan asalnya (misalnya, Swiss dan Denmark). Sebaliknya, kontribusi dengan tanda negatif terjadi ketika negara berada di atas rata-rata satu variabel dan di bawah rata-rata di sisi lain (misalnya, Swedia dan Amerika Serikat)

Tabel 9.3. Kontribusi Kasus-Spesifik untuk Perkiraan Koefisien.

	Denmark	**Japan**	**Sweden**	**United States**	**Lainnya**	**Total**
Model 1						
Ketimpangan	0.09	-0.22	-0.08	-0.17	0.03	-0.36
Model 2						
Kemiskinan	0.12	0.06	-0.07	-0.16	0.13	0.07
Model 3						
Ketimpangan	0.02	-0.47	-0.06	-0.12	-0.10	-0.73
Kemiskinan	0.10	0.38	-0.03	-0.08	0.19	0.56

Model 4						
Ketimpangan	0.04	-0.45	-0.07	-0.09	-0.10	-0.68
Kemiskinan	0.15	0.45	-0.07	0.01	0.21	0.75
Interaksi	-0.12	-0.19	0.09	-0.24	-0.04	-0.49

Apa yang dikatakan intensitas tentang efek ketidaksetaraan pada harapan hidup? Kita lihat dari Gambar 2 bahwa efek ini disebabkan terutama oleh Jepang (−0.2209), Amerika Serikat (−0.1724), Denmark (0.0896), dan Swedia (−0.0795). 16 negara-negara yang tersisa berkontribusi sedikit secara agregat, hanya jumlah 0,0275 dengan koefisien regresi keseluruhan untuk ketidaksetaraan. (Perhatikan bahwa kelima angka ini menjumlahkan semua koefisien regresi −0.36, sebagaimana mestinya.) Arah intensitas ini mencerminkan kombinasi prediktor dan arah hasil: Denmark memiliki ketimpangan yang rendah dan harapan hidup rendah (asosiasi positif); Jepang dan Swedia memiliki ketimpangan yang rendah dan harapan hidup yang tinggi, sedangkan Amerika Serikat memiliki ketimpangan yang tinggi dan harapan hidup yang rendah (arah negatif).

Nilai *p* untuk koefisien regresi (−0.36) pada Gambar 2 lebih besar dari 0,05 (lihat model 1 pada Tabel 2). Pemeriksaan dari Gambar 2 menunjukkan bahwa dari empat negara dengan kontribusi yang terbesar terhadap koefisien regresi keseluruhan, Denmark adalah *outlier* terbesar sehubungan dengan garis pas (memiliki residu terbesar). Memang, jika bangsa yang satu ini dihilangkan, maka koefisien regresi di 19 negara lainnya menjadi −0.47 (p = 0.02 pada 18 derajat kebebasan). Gambar 3 menunjukkan pengaruh kemiskinan terhadap harapan hidup di model 2 (+0.07, p > 0.05). Meskipun efeknya positif, tetapi

kontribusi unik terbesar negatif: intensitas Amerika Serikat adalah −0.16 (kemiskinan tinggi, harapan hidup rendah). Denmark mengikuti dengan intensitas +0.12 karena kombinasi kemiskinan yang rendah dan harapan hidup yang rendah. Swedia memiliki intensitas terbesar ketiga (−0.07; kemiskinan rendah dan harapan hidup tinggi). Israel dan Jepang mengikuti, masing-masing dengan intensitas sekitar +0.06. Kedua negara tersebut memiliki tingkat kemiskinan yang relatif tinggi (khususnya Israel) dan harapan hidup yang tinggi (khususnya Jepang).

Koefisien regresi memiliki nilai *p* yang lebih besar dari 0.05, tetapi untuk alasan yang berbeda dari pada Gambar 2, di mana Denmark dipandang sebagai *outlier*. Pada Gambar 3, lima negara dengan kontribusi besaran tertinggi dibatalkan satu sama lain: jumlah kontribusi Amerika Serikat, Denmark, Swedia, Israel, dan Jepang hanya +.0024. 15 negara lainnya yang kontribusinya relatif lebih rendah menghasilkan total kontribusi +0.0633 untuk koefisien regresi total +.0657 (dibulatkan menjadi +0.07 dalam model 2 dari Tabel 2). Nyatanya, Gambar 3 mengilustrasikan poin tematik penting kita: kita harus mencari bukan untuk *outlier* tunggal, tetapi untuk keseluruhan pola kontribusi kasus karena koefisien regresi hanyalah jumlah di seluruh kontribusi berbasis kasus ini.

Efek terurai dari ketimpangan (dalam Gambar 4A: koefisien = −0.73, $p = 0.016$) dan kemiskinan (dalam Gambar 4B: koefisien = +0.56, $p = 0.051$) pada angka harapan hidup. Jepang dan Amerika Serikat memiliki intensitas terbesar untuk ketimpangan pendapatan: −0.47 (ketimpangan rendah dan harapan hidup tinggi) dan −0.12 (ketimpangan tinggi dan harapan hidup renda h). Jepang juga mendominasi koefisien kemiskinan dengan intensitas +0.38 (tingkat kemiskinan yang relatif tinggi dan harapan hidup tinggi) diikuti oleh Denmark (+0.10). Angka 4 mengilustrasikan bagaimana metode penulis untuk menginterpretasikan kontribusi meluas ke regresi

berganda: alih-alih mengambil ketidaksetaraan (misalnya) sebagai sumbu x, penulis menggunakan residu dari regresi variabel ini pada semua prediktor lainnya. (Pada Gambar 4A hanya prediktor lainnya adalah kemiskinan.) Dengan ketentuan ini, daerah persegi panjang persis sebanding dengan kontribusi berbasis kasus. Akhirnya, penulis beralih ke model 4. Model regresi berganda termasuk istilah interaksi (koefisien untuk ketimpangan adalah −0.68, $p = 0.14$; koefisien untuk kemiskinan adalah +0.75, $p = .010$; dan koefisien interaksinya adalah −0.49, $p = .031$).

Jelas bahwa efek interaksi disebabkan secara dominan kontribusi dari Amerika Serikat, Jepang, Denmark, dan Swedia; jumlah kontribusi mereka pada Gambar 5 (lihat juga Tabel 3) adalah −0.451, dari total koefisien regresi dari −0.488. Keempat negara ini, kemudian, mendominasi kontribusi spesifik kasus terhadap model koefisien regresi di setiap negara yang memperkirakan pengaruh ketimpangan pendapatan, kemiskinan—secara individual atau secara bersamaan dipertimbangkan—dan interaksi mereka pada harapan hidup.

Penulis mencatat sebelumnya bahwa berbagai cerita mungkin mendasari analisis regresi dan temuan penulis menunjukkan bahwa proposisi ini benar apakah koefisien mencapai "statistik". signifikansi" (secara konvensional diwakili oleh nilai $p < 0,05$) atau tidak. Terlepas dari kecocokan model atau variasi yang dijelaskan, cerita di balik asosiasi ini sepertinya selalu ada berputar di sekitar empat negara yang sama. Jadi, satu pertanyaan muncul: mengapa negara-negara ini? Apa yang menjadi ciri mereka?

Investigasi data yang tidak biasa dan berpengaruh, berdasarkan jarak Cook dan DFFITS (Belsley, Kuh, dan Welsch 2004; Chen dkk. 2003), menunjukkan bahwa Amerika Serikat, Jepang, Denmark, dan Swedia adalah kasus yang sangat berpengaruh. Kasus bisa menjadi tidak biasa dalam tiga cara.

Outlier memiliki residu yang besar yang berarti bahwa hasil mereka diprediksi dengan buruk oleh persamaan regresi. Jika *outlier* dihapus dari analisis, kecocokan model meningkat.

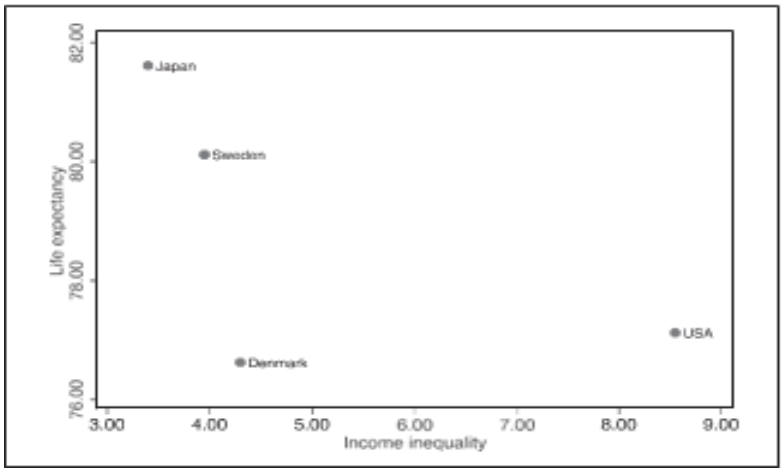
Kasus memiliki *leverage* yang tinggi jika nilainya pada prediktor tidak biasa; kasus ini dapat mempengaruhi estimasi koefisien regresi (seperti Amerika Serikat pada Gambar 5). Akhirnya, kasusnya berpengaruh tinggi yang dapat dianggap sebagai produk dari *outlierness* dan *leverage* jika penghilangan mereka secara drastis mengubah perkiraan koefisien. Jarak Cook dan DFFITS adalah ukuran pengaruh: kasus di atas ambang batas tertentu sangat berpengaruh. Amerika Serikat, Jepang, Denmark, dan Swedia memiliki pengaruh tinggi pada kedua ukuran (lihat Tabel S4 dan S5 di data tambahan). Menarik juga untuk membandingkan kontribusi spesifik negara dengan DFBETAS yang memperkirakan bagaimana menjatuhkan setiap kasus mengubah koefisien regresi (Chen et al. 2003). Di keempat model, DFBETAS adalah sangat berkorelasi dengan intensitas yang penulis hitung dengan satu pengecualian utama: istilah interaksi. Dalam hal ini, koefisien korelasi antara intensitas dan DFBETAS adalah 0,24 (p = 0,307). Kurangnya korelasi ini terutama disebabkan oleh empat kasus biasa: Amerika Serikat, Jepang, Denmark, dan Swedia (lihat Gambar S2 di data tambahan).

Hubungan antara intensitas dan DFBETAS layak mendapat perhatian di masa depan mengingat kedekatan konseptual mereka dan korelasi empiris yang sering penulis amati. Penulis menyampaikan informasi intensitas dan DFBETAS yang terkait, tetapi berbeda. DFBETAS menunjukkan seberapa besar perubahan koefisien regresi sebagai akibat dari perubahan dalam ruang properti (perubahan model) yang terjadi saat kita menjatuhkan kasus individu. Berbeda dari ini, intensitas kasus penulis menunjukkan seberapa besar kontribusi masing-masing kasus terhadap suatu koefisien untuk ruang properti tertentu (yaitu, untuk model tertentu). Dan sebagai

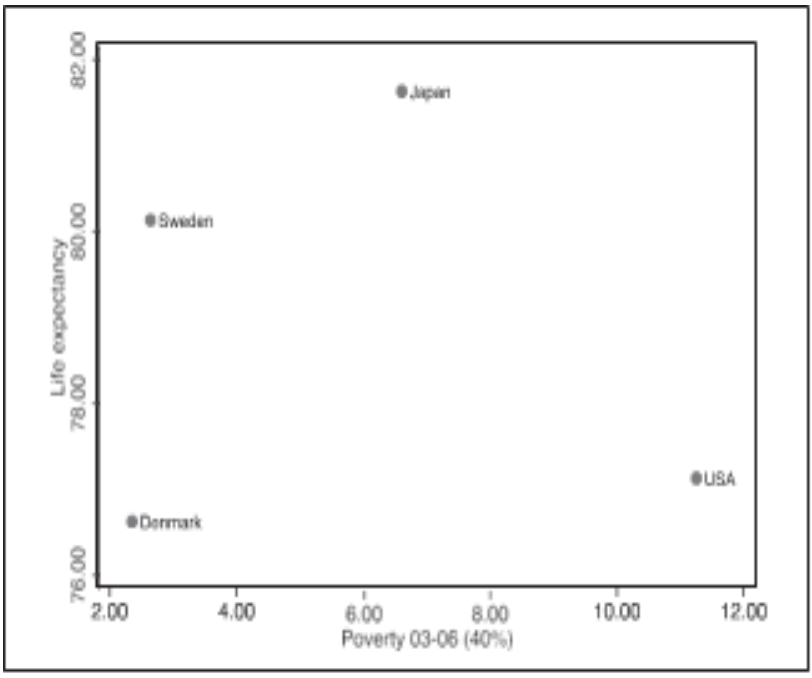
penulis tunjukkan, intensitas memberikan dekomposisi masing-masing koefisien regresi.

9.6 Mempelajari dari Kasus

Apa yang bisa kita pelajari dari empat kasus berpengaruh ini? Satu cara sederhana untuk mencoba jawaban adalah dengan melihat negara-negara ini aktif dalam *scatterplot* yang membandingkan harapan hidup dengan ketimpangan pendapatan (Gambar 6) dan kemiskinan (Gambar 7).



Gambar 6: Harapan hidup dengan ketimpangan pendapatan: kasus berpengaruh.



Gambar 7: Harapan hidup berdasarkan kemiskinan: kasus berpengaruh.

Para sarjana menunjukkan bahwa kesehatan yang lebih baik dikaitkan dengan ketimpangan pendapatan yang lebih rendah (Wilkinson dan Pickett 2009) dan (atau) kemiskinan yang lebih rendah (Ludwig et al. 2011, 2012). Gambar 6 dan 7 menunjukkan bahwa hanya dua negara yang memenuhi kedua harapan teoretis tersebut. Amerika Serikat menunjukkan tingkat ketimpangan yang tinggi dan kemiskinan serta harapan hidup yang sangat rendah. Sebaliknya, Swedia menunjukkan ketimpangan pendapatan dan tingkat kemiskinan yang rendah dan umur panjang yang tinggi. Sebaliknya, kasus Denmark dan Jepang menantang harapan teoretis penulis. Denmark punya harapan hidup yang sangat rendah (lebih rendah dari Amerika Serikat) meskipun memiliki ketimpangan yang rendah dan kemiskinan yang rendah. Jepang punya harapan hidup yang tinggi meskipun memiliki tingkat kemiskinan yang relatif tinggi

yang berada di suatu tempat antara negara-negara Skandinavia dan Amerika Serikat.

Denmark adalah bagian dari model Nordik yang sangat dihormati adalah rezim kesejahteraan sosial demokrat (Esping-Andersen 1990) yang unggul dalam berbagai hasil sosial. Di Denmark, harapan hidup, bagaimanapun, bukanlah salah satu dari hasil ini. Kesenjangan umur panjang Denmark cukup mengesankan berjumlah 3,5 tahun lebih sedikit dari tetangga dekatnya, Swedia. Untuk menempatkan ini ke dalam perspektif, orang harus ingat bahwa di antara set dari 20 negara kaya, 1,1 tahun sesuai dengan 1 standar deviasi, yang berarti harapan hidup Denmark 3 standar deviasi lebih rendah dari Swedia. Umur panjang yang rendah di antara orang Denmark didokumentasikan dengan baik di literatur kesehatan. Penjelasan yang mungkin adalah melihat peran ketimpangan kekayaan (Nowatzki 2012), khususnya tinggi di Denmark. Namun, sejumlah analisis mengidentifikasi sebuah kontribusi yang lebih spesifik untuk umur panjang yang rendah: risiko tinggi untuk kematian bagi wanita yang lahir di antara dua perang dunia (Jacobsen et al. 2004; Jacobsen, Keiding, dan Lynge 2002; Juli 2008; Juel, Bjerregaard, dan Madsen 2000; Lindahl Jacobsen dkk. 2016). Wanita Denmark antara dua perang dunia "dipamerkan" tingginya tingkat merokok sepanjang perjalanan hidup mereka dan merokok lebih banyak daripada wanita di negara Eropa lainnya (Bukit 1992).

Tidak jelas mengapa ini terjadi, tetapi beberapa analis memiliki usulan penjelasan budaya berdasarkan pekerjaan: Wanita Denmark memasuki pasar tenaga kerja lebih awal dari wanita di sebagian besar negara dan otonomi sosial ekonomi yang diakibatkannya mungkin telah membuat mereka menganggap merokok sebagai hal yang sah perilaku (Juel et al. 2000; Nathanson 1995). Denmark punya juga kebijakan tembakau yang agak liberal: merokok dilarang di restoran, dengan pengecualian, hanya pada tahun 2007 (Christensen et al.

2010). Selain perilaku spesifik gender ini, orang Denmark mengonsumsi lebih banyak lemak daripada negara Nordik lainnya (Juel et al. 2000). Kombinasi dari semua faktor ini mungkin terjadi bertanggung jawab atas rendahnya harapan hidup Denmark. Namun, harapan hidup saat ini sedang meningkat lagi. Faktanya, saat wanita antar perang (lahir dari tahun 1915 hingga 1945) meninggal dunia, Denmark mengalami peningkatan umur panjang yang lebih tajam daripada negara Nordik lainnya (Lindahl-Jacobsen et al. 2016): misalnya temuan mendukung gagasan bahwa efek kohort benar-benar bertanggung jawab atas umur panjang yang diamati di Denmark.

Pada tahun 1947, harapan hidup di Jepang adalah sekitar 20 tahun lebih rendah daripada di Swedia dan sebagian besar negara Eropa Barat; pada tahun 1990, orang Jepang hidup lebih lama dari siapa pun (Matsuzaki 1992). Beberapa peneliti mencoba untuk menjelaskan umur panjang Jepang dengan argumen sosiokultural, seperti dukungan sosial yang kuat, kehidupan yang serba lambat, pemujaan leluhur, penghormatan terhadap orang tua, dan sentralitas keluarga secara keseluruhan, terutama untuk daerah-daerah yang memiliki angka harapan hidup tinggi meskipun memiliki peringkat indikator sosial ekonomi yang rendah, seperti seperti Okinawa (Cockerham, Hattori, dan Yamori 2000). Hipotesis bahwa kain yang erat dapat melindungi komunitas kesehatan dan bahkan mengimbangi faktor risiko yang khas sudah bukan epidemologi social yang asing lagi. Salah satu contoh yang menonjol adalah Roseto efek (Bruhn et al. 1966; Egolf et al. 1992), yang diajukan untuk menjelaskan mengapa penduduk Roseto Italia-Amerika, Pennsylvania, meskipun mengalami tingkat penyakit jantung yang rendah gaya hidup mereka yang tidak sehat. Namun, umur panjang Jepang tidak dapat sepenuhnya dipahami tanpa mempertimbangkan perubahan kebiasaan diet yang terjadi selama abad kedua puluh.

Diet orang Jepang memiliki sejumlah karakteristik manfaat harapan hidup yang panjang dan sehat. Beras adalah hidangan yang utama, asupan ikan sangat tinggi, dan rasio protein hewani terhadap sayuran sangat seimbang (Matsuzaki 1992). Konsumsi teh hijau yang sangat populer di Jepang dikaitkan dengan penurunan angka kematian dari semua penyebab dan dari penyakit kardiovaskular (Kokubo et al. 2013; Kuriyama dkk. 2006). Terakhir, penurunan tajam asupan garam setelah tahun 1950-an dikaitkan dengan penurunan tinggi tekanan darah dan stroke (Miura 2011).

9.7 Penutup

Dalam penelitian ini, penulis menggunakan metodologi baru (Breiger et al. 2011; Breiger dan Melamed 2014; Melamed, Breiger, dkk. 2012; Melamed, Schoon, dkk. 2012) yang mengubah model regresi luar dalam. Dengan pendekatan ini, penulis dapat memdekomposi koefisien regresi menjadi kontribusi spesifik negara yang menunjukkan intensitas setiap kasus pada perkiraan efek dan penulis dapat mengungkap berbagai dinamika analisis regresi yang mendasarinya.

Penulis menerapkan metodologi ini pada studi perbandingan kesehatan dan ketidaksetaraan. Penulis menunjukkan bahwa logika yang sama dari kasus kontribusi mendasari hubungan terlepas dari kecocokan model. Sambil menilai harapan hidup apa yang lebih penting — ketimpangan, kemiskinan, atau interaksinya — orang mungkin gagal memperhatikan bahwa empat kasus yang sama sangat berpengaruh (Amerika Serikat, Swedia, Denmark, dan Jepang) mendominasi semua model. Setiap analisis makro-komparatif dari determinan sosial kesehatan populasi ditakdirkan untuk meleset kecuali logika kontribusi kasus terungkap. Dengan memutar regresi "dalam ke luar", penulis membawa logika kontribusi kasus ke pusat analisis.

Bab 10

Learning

10.1 Pengantar Machine Learning

Machine Learning merupakan bagian Kecerdasan buatan (Artificial Inteligences/AI). Pengertian Machine Learning dari sudut pandang konseptual merupakan suatu proses untuk meningkatkan pengetahuan program AI. Tujuan dari machine learning ini adalah untuk memahami struktur data dan memasukan data tersebut ke dalam model yang dapat dipahami dan dimamfaatkan oleh orang-orang. Setiap pengguna teknologi saat ini sudah memanfaatkan machine learning contohnya teknologi wajah memungkinkan platform media social yang dapat membantu pengguna menandai dan berbagi foto dan contoh lainnya adalah mobil self-driving.

Secara konseptual Learning adalah proses yang meningkatkan pengetahuan program AI dengan melakukan pengamatan tentang lingkungannya. Sementara secara matematis/teknis Learning merupakan proses pembelajaran AI berfokus pada pemrosesan kumpulan pasangan input-output untuk fungsi tertentu dan memprediksi output untuk input baru. Terdapat dua kelompok utama model dalam Learning yaitu *supervised* dan *unsupervised*.

Pengetahuan dan Umpan balik dapat dijadikan sebagai model untuk memudahkan dalam mempelajari Learning model, dimana dalam perspektif pengetahuan Learning model dapat

direpresntasikan melalui titik data input dan ouput. Dan dalam umpan balik learning model dapat diklasifiksikanberdasarkan interaksi dengan lingkungan luar,pengguna dan factor eksternal lainnya.

10.2 AI Learning Model : Knowledge-Based Cassification

Sebagai representasi dari pengetahuan maka AI learning model dapat diklasifikan menjadi 2 tipe utama yaitu : inductive dan deductive

a. Inductive learning

   Model Learning Inductive merupakan jenis model Learning AI yang didasarkan pada aturan umum dari kumpulan pasangan input –output. Algoritma seperti Knowledge Based Inductive Learning (KBIL) merupakan salah satu contoh yang aling bagus untuk model learning ini dimana algoritma KBIL ini berfokus pada menemukan hipotesis induktif pad akumpulan data dengan bantuan informasi latar belakang

b. Deductive learning

   Deductive learning merupakan jenis teknik learning Ai yang dimulai dengan serangkaian aturan dan menyimpulkan adanya aturan baru yanglebih efisien dalam algoritma AI tertentu. Explanation Learning Based (EBL) dan Relevance-0 Based Learning (RBL) merupakan contoh dari teknik deduktif. EBL mengekstrak aturan umum dari contoh dengan "**menggeneralisasi**" penjelasannya sedangkan RBL berfokus pada identifikasi atribut dan "**generalisasi**" deduktif dari contoh yang sederhana.

10.3 AI Learning Models: Feedback-Based Classification

Berdasarkan karakteristik dari feedback, AI learning models dapat diklasifikasikan menjadi supervised, unsupervised, semi supervised or reinforced.

A. Supervised learning

Supervised learning merupakan salah satu pendekatan machine yang ditentukan oleh pengguna kumpulan data berlabel yang digunakan melatih algoritma untuk mengkalsifaikasi data dan memperdiksa hasil (output). Dataset berlabel memliki keluaran yang sesuai dengan data input agar mesin memahami apa yang harus dicari dalam data yang tidak terlihat.

Gambar 1: Supervised Learning

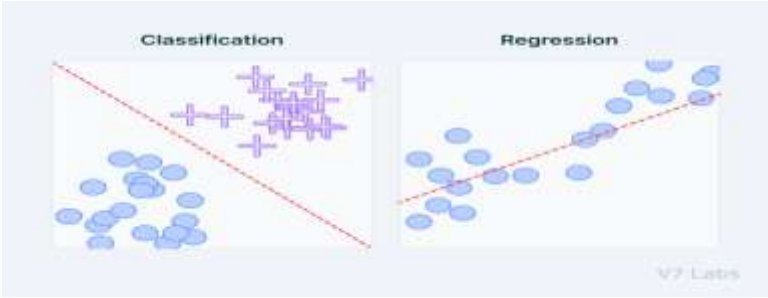
Terdapat 2 area utama pada supervised learning yaitu :

1. Classification

   Clasification mengacu pada mengambilnilai input dan memetakannya ke nilai diskrit. Dalam malasalah Clasification output biasanya teridi dari kelas atau kategori. Conothnya memprediksi warna sebuah object atau memprediksi hari ini hujan atau tidak hujan

2. Regression

   Regression berhubungan dengan data continue (fungsi nilai). Dalam regresi, nilai keluaran yang diprediksi adalah bilangan real. Contohnya : memprediksi harga rumah,tren saham pada waktu tertentu

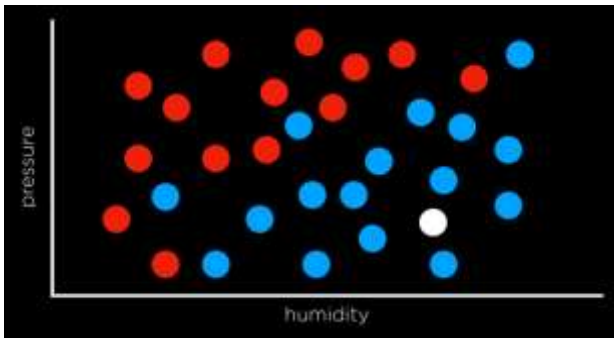


Gambar 2. Perbedaan Klasifikasi Dan Regresi

1. Clasification

Contoh berikut ini merupakan untuk supervised learning dimana pada hal ini metode yang digunakan adalah classification. Ini adalah tugas di mana fungsi memetakan input ke output diskrit. Sebagai contoh, diberikan beberapa informasi tentang kelembaban dan tekanan udara untuk hari tertentu (input), komputer memutuskan apakah hari itu akan hujan atau tidak (output). Komputer melakukan ini setelahmelakukan training data pada kumpulan data dengan beberapa hari di mana kelembaban dan tekanan udara sudah dipetakan apakah hujan atau tidak.

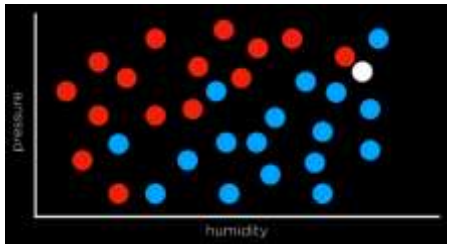
Tugas ini dapat diformalkan sebagai berikut: Ketika mengamati alam, di mana fungsi *f(kelembaban, tekanan)* memetakan input ke nilai diskrit, baik Hujan atau Tanpa Hujan. Fungsi ini kemudian disembunyikan, dan mungkin dipengaruhi oleh banyak variabel lain yang tidak dapat di akses. Tujuannya adalah untuk membuat fungsi *h(kelembaban, tekanan)* yang dapat mendekati perilaku *fungsi f*. Tugas tersebut dapat divisualisasikan dengan memplot hari pada dimensi kelembaban dan hujan (input), mewarnai setiap titik data dengan warna biru jika hari itu hujan dan merah jika hari itu tidak hujan (output). Titik data putih hanya memiliki input, dan komputer perlu mengetahui outputnya.



Gambar 3.Penggambaran Untuk Kelembatan Udara

a. Nearest-Neighbor Classification

Salah satu cara untuk menyelesaikan tugas seperti yang dijelaskan di atas adalah dengan menetapkan variabel yang bersangkutan nilai pengamatan terdekat. Jadi, misalnya, titik putih pada grafik di atas akan berwarna biru, karena titik terdekat yang diamati juga berwarna biru. Ini mungkin bekerja dengan baik beberapa kali, tetapi pertimbangkan grafik di bawah ini.



Gambar 4. Nearest neighborhood classification

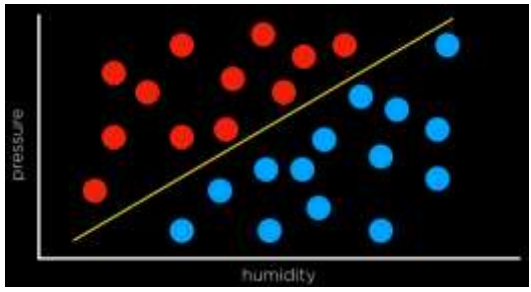
Mengikuti strategi yang sama, titik putih harus diwarnai merah, karena pengamatan terdekatnya juga berwarna merah. Namun, melihat gambaran yang lebih besar, sepertinya sebagian besar pengamatan lain di sekitarnya berwarna biru, yang mungkin memberi kita intuisi bahwa biru adalah prediksi yang lebih baik dalam kasus ini, meskipun pengamatan terdekat berwarna merah.

Salah satu cara untuk mendapat batasan klasifikasi tetangga terdekat adalah dengan menggunakan *k-nearest-neighbors classification* dimana titik diwarnai berdasarkan warna yang paling sering dari k tetangga terdekat. Terserah programmer untuk memutuskan apa itu k. Menggunakan klasifikasi 3-tetangga terdekat, misalnya, titik putih di atas akan berwarna biru, yang secara intuitif tampak seperti keputusan yang lebih baik.

Kelemahan dari klasifikasi k-nearest-neighbors adalah bahwa, menggunakan pendekatan naif, algoritma harus mengukur jarak setiap titik ke titik yang bersangkutan, yang secara komputasi akan susah didapatkan. Ini dapat dipercepat dengan menggunakan struktur data yang memungkinkan menemukan tetangga lebih cepat atau dengan memangkas pengamatan yang tidak relevan.

b. Perceptron Learning

Cara lain untuk mengatasi masalah klasifikasi, yang bertentangan dengan nearest neighbor terdekat, adalah melihat data secara keseluruhan dan mencoba membuat batas keputusan. Dalam data dua dimensi, kita dapat menarik garis antara dua jenis pengamatan. Setiap titik data tambahan akan diklasifikasikan berdasarkan sisi garis yang diplot.



Gambar 5 Perceptron Learning

Kelemahan dari pendekatan ini adalah datanya berantakan, dan jarang seseorang dapat menarik garis dan membagi kelas dengan rapi menjadi dua pengamatan tanpa kesalahan. benar lebih sering daripada tidak, tetapi terkadang masih salah mengklasifikasikannya.

Seringkali, kita akan melakukan dengan memberikan batas yang memisahkan pengamatan dengan dengan benar , tetapi terkadang masih salah dalam mengklasifikasikannya.

Dalam hal ini inputnya berupa :

- x_1 = Humidity
- x_2 = Pressure

dengan memberikan hypothesis fungsi (x1,x2), maka output akan memprediksi cuaca apakah akan hujan atau tidak. Hal ini dapat dilakukan dengan melakukan pengecekan terhadap batasan dari observasi yang dilakukan, yang akhirnya akan menggunakan persamaan linear seperti berikut ini :

- Rain $w_0 + w_1x_1 + w_2x_2 \geq 0$
- No Rain otherwise

Seringkali keluaran dari persamaan ini akan dikodekan menjadi 1 dan 0, dimana jika persamaan menghasilkan lebih dari 0, outputnya adalah 1 (Hujan), dan 0 sebaliknya (Tidak Ada Hujan). Bobot dan nilai diwakili oleh vektor, yang merupakan urutan angka (yang dapat disimpan dalam daftar atau tupel dengan Python). Kami menghasilkan Vektor Bobot *w: (w_0, w_1, w_2)*, dan mendapatkan vektor bobot terbaik adalah tujuan dari algoritma learning ini dan akan menghasilkan Vektor Input x: (1, x_1, x_2). Dengan mengambil produk titik dari dua vektor. Artinya, kita mengalikan setiap nilai dalam satu vektor dengan nilai yang sesuai dalam vektor kedua, sampai pada ekspresi di atas: $w_0 + w_1x_1 + w_2x_2$. Nilai pertama dalam vektor input adalah 1

karena, ketika dikalikan dengan vektor bobot w_0, kita ingin membuatnya konstan.

Hipotesis yang kita gunakan dapat di representasikan melalui fungsi berikut ini :

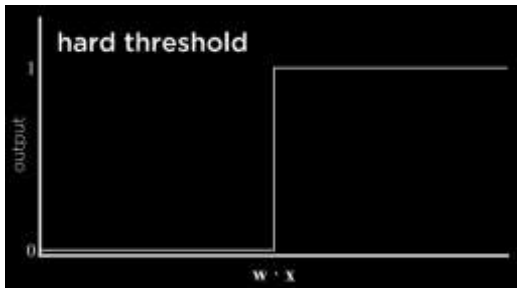
$$h_{\mathbf{w}}(\mathbf{x}) = \begin{matrix} 1 \text{ if } \mathbf{w} \cdot \mathbf{x} \geq 0 \\ 0 \text{ otherwise} \end{matrix}$$

Karena tujuan dari algoritma adalah untuk menemukan vektor bobot terbaik, ketika algoritma menemukan data baru, ia memperbarui bobot. Ia melakukannya dengan menggunakan aturan perceptron learning.

$$w_i = w_i + \alpha(y - h_{\mathbf{w}}(\mathbf{x})) \times x_i$$

Hal penting yang dapat diambil dari aturan ini adalah bahwa untuk setiap titik data,dialukan penyesuain bobot untuk membuat fungsi menjadi lebih akurat. Detailnya adalah bahwa setiap bobot diatur sama dengan dirinya sendiri ditambah beberapa nilai nilai dalam tanda kurung. Di sini, y adalah singkatan dari nilai yang diamati sedangkan fungsi hipotesis adalah perkiraan. Jika mereka identik, seluruh istilah ini sama dengan nol, dan dengan demikian bobotnya tidak berubah. Tetapi jika disebut (Tidak Ada Hujan saat Hujan diamati), maka nilai dalam tanda kurung akan menjadi 1 dan bobot akan bertambah dengan nilai x_i yang diskalakan oleh koefisien Learning. Jika kita melebih-lebihkan (memanggil Hujan saat Tidak Ada Hujan diamati), maka nilai dalam kurung akan menjadi -1 dan bobot akan berkurang dengan nilai x yang diskalakan oleh . Semakin tinggi , semakin kuat pengaruh setiap peristiwa baru terhadap bobot.

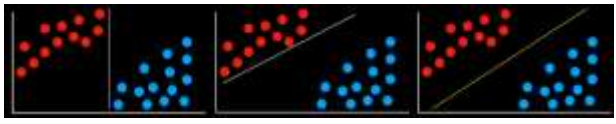
Hasil dari proses ini adalah fungsi ambang batas yang beralih dari 0 ke 1 setelah nilai perkiraan melewati beberapa ambang batas.



Gambar 6. Hard threshold

Masalah dengan jenis fungsi yang menggunakan ambang batas ini adalah fungsi ini tidak dapat mengungkapkan ketidakpastian, karena hanya bisa sama dengan 0 atau 1. Fungsi ini menggunakan ambang batas yang keras. Cara untuk menyiasatinya adalah dengan menggunakan fungsi logistik, yang menggunakan ambang batas lunak. Fungsi logistik dapat menghasilkan bilangan real antara 0 dan 1, yang akan menyatakan keyakinan dalam estimasi. Semakin mendekati nilai 1, semakin besar kemungkinan hujan.

Selain tetangga terdekat dan regresi linier, pendekatan lain untuk klasifikasi adalah Support Vector Machine (SVM) . Pendekatan ini menggunakan vektor tambahan (vektor pendukung) di dekat batas keputusan untuk membuat keputusan terbaik saat memisahkan data. Perhatikan contoh di bawah ini.

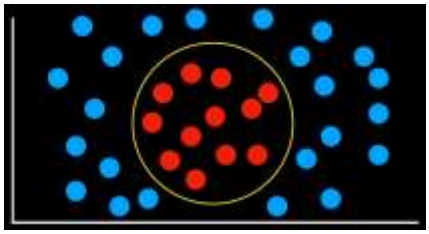


Gambar 8. Support Vector Machines

Semua batasan berfungsi karena memisahkan data tanpa kesalahan. Namun, apakah mereka sama baiknya? Dua batas keputusan paling kiri sangat dekat dengan beberapa pengamatan. Ini berarti bahwa titik data baru yang hanya sedikit berbeda dari satu kelompok dapat salah diklasifikasikan sebagai yang lain. Berlawanan dengan itu, batas keputusan

paling kanan menjaga jarak paling jauh dari masing-masing kelompok, sehingga memberikan kelonggaran paling banyak untuk variasi di dalamnya. Jenis batas ini, yang sejauh mungkin dari dua kelompok yang dipisahkannya, disebut Pemisah Margin Maksimum.

Manfaat lain dari Support Vector Machines adalah mereka dapat mewakili batas-batas keputusan dengan lebih dari dua dimensi, serta batas-batas keputusan non-linear, seperti di bawah ini.



Gambar 9. Support Vector Machines

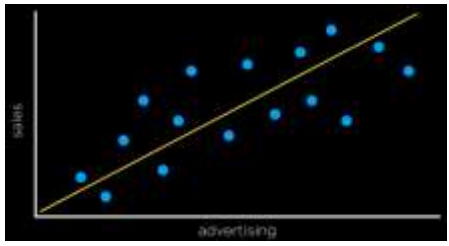
Berdasarkan penjelasan daitas terdapat beberapa cara untuk menyelesaikan masalah Clasification dengan tidak ada yang selalu lebih baik dari yang lain. Masing-masing memiliki kekurangan dan mungkin terbukti lebih berguna daripada yang lain dalam situasi tertentu.

2. Regression

Regresi adalah supervised learning yang bertugas sebagai sebuah fungsi yang dapat memetakan titik input ke nilai kontinu, beberapa bilangan real. Ini berbeda dari klasifikasi di mana masalah klasifikasi memetakan input ke nilai diskrit (Hujan atau Tanpa Hujan).

Misalnya, sebuah perusahaan mungkin menggunakan regresi untuk menjawab pertanyaan tentang bagaimana uang yang dihabiskan untuk iklan memprediksi uang yang diperoleh dari penjualan. Dalam hal ini, fungsi yang diamati *f(periklanan)* mewakili pendapatan yang diamati setelah sejumlah uang yang

dihabiskan untuk iklan (perhatikan bahwa fungsi tersebut dapat mengambil lebih dari satu variabel input). Ini adalah data yang kami mulai. Dengan data ini, kami ingin membuat hipotesis fungsi *h*(*iklan*) yang akan mencoba mendekati perilaku *fungsi f*. *h* akan menghasilkan garis yang tujuannya bukan untuk memisahkan antara jenis pengamatan, tetapi untuk memprediksi, berdasarkan input, apa yang akan menjadi nilai output.



Gambar 10. Regression

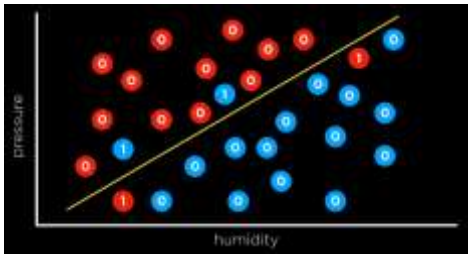
a. Loss Functions

Loss Function adalah cara untuk mengukur utilitas yang hilang oleh salah satu aturan keputusan di atas. Semakin kurang akurat prediksinya, semakin besar kerugiannya.

Untuk masalah klasifikasi, kita dapat menggunakan 0-1 Loss Function.

- L(aktual, diprediksi):
  - 0 jika aktual = diprediksi
  - 1 sebaliknya

Dengan kata lain, fungsi ini memperoleh nilai ketika prediksi tidak benar dan tidak mendapatkan nilai ketika benar (yaitu ketika nilai yang diamati dan diprediksi cocok).



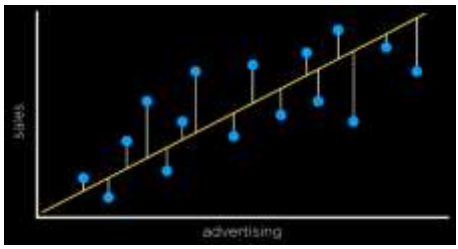
Gambar 11. Loss Function

Pada contoh di atas, hari-hari yang bernilai 0 adalah hari-hari di mana kita memprediksi cuaca dengan benar (hari hujan di bawah garis dan bukan hari hujan di atas garis). Namun, hari-hari ketika tidak hujan di bawah garis dan hari-hari ketika hujan di atasnya adalah hari-hari yang gagal diprediksi. Kami memberikan masing-masing nilai 1 dan menjumlahkannya untuk mendapatkan perkiraan empiris tentang seberapa lossy batas keputusan kami.

L_1 dan L_2 merupakan loss function yang dapat digunakan saat memprediksi nilai kontinu. Dalam hal ini untuk mengukur untuk setiap prediksi seberapa besar perbedaannya dari nilai yang diamati. Maka dilakukan dengan mengambil nilai absolut atau nilai kuadrat dari nilai yang diamati dikurangi nilai yang diprediksi (yaitu seberapa jauh prediksi dari nilai yang diamati).

- L_1: L(aktual, diprediksi) = |aktual - diprediksi|
- L_2: L(aktual, prediksi) = (aktual - prediksi)^2

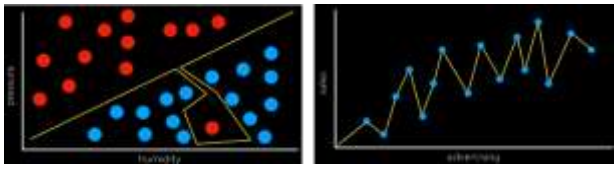
Seseorang dapat memilih fungsi kerugian yang paling sesuai dengan tujuan mereka. L_2 menghukum outlier lebih keras daripada L_1 karena menguadratkan perbedaan. L_1 dapat divisualisasikan dengan menjumlahkan jarak dari setiap titik yang diamati ke titik yang diprediksi pada garis regresi:



Gambar 12. Loss function

b. Overfitting

Overfitting adalah ketika model sangat cocok dengan data pelatihan sehingga gagal digeneralisasi ke kumpulan data lainnya. Dalam hal ini, fungsi kerugian adalah pedang bermata dua. Dalam dua contoh di bawah ini, fungsi kerugian diminimalkan sedemikian rupa sehingga kerugiannya sama dengan 0. Namun, kecil kemungkinannya akan cocok dengan data baru dengan baik.



Misalnya, di grafik kiri, titik di sebelah merah di bagian bawah layar kemungkinan besar adalah Hujan (biru). Namun, dengan model overfitted, akan diklasifikasikan sebagai No Rain (merah).

a. Regularization

Regularization adalah proses muntuk mendapatkan hipotesis yang lebih kompleks untuk mendukung hipotesis yang lebih sederhana dan lebih umum. Regularization biasanya digunakan untuk menghindari overfitting.

Dalam regulazitaion, diperkirakan biaya fungsi hipotesis h dengan menjumlahkan kerugiannya dan ukuran kompleksitasnya.

biaya(h) = kerugian(h) + kompleksitas(h)

Lambda (λ) adalah konstanta yang dapat kita gunakan untuk memodulasi seberapa kuat untuk menghukum kompleksitas dalam fungsi biaya kita. Semakin tinggi , semakin mahal kompleksitasnya.

Salah satu cara untuk menguji apakah kita melakukan *overfitted model* adalah dengan *Holdout Cross Validation.* Dalam teknik ini, semua data dibagi menjadi dua*: set pelatihan* dan *set pengujian*. Dnegan menggunakan learning algortima pada set pelatihan, dan kemudian melihat seberapa baik prediksi data dalam set pengujian. Idenya di sini adalah bahwa dengan menguji data yang tidak digunakan dalam pelatihan, kita dapat mengukur seberapa baik pembelajaran digeneralisasikan.

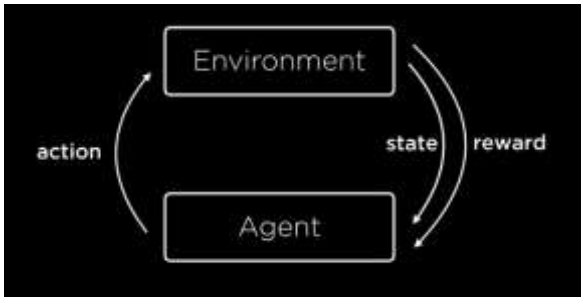
Kelemahan dari validasi silang ketidaksepakatan adalah tidak bisa melatih model pada setengah data, karena digunakan untuk tujuan evaluasi. Cara untuk mengatasinya adalah dengan menggunakan k-Fold Cross-Validation. Dalam proses ini, data dibagi menjadi k set. Kemuadian dilakukan training untuk k setiap training terhadap k akan meninggalkan satu set data dan menggunakannya sebagai set pengujian. Hal ini akan berakhir dengan k evaluasi berbeda dari model yang dapat di rata-rata dan mendapatkan perkiraan tentang bagaimana model digeneralisasi tanpa kehilangan data apa pun.

b. *Scikit-learn*

Pada Python, ada beberapa library yang memungkinkan kita untuk menggunakan algoritme machine learning dengan nyaman. Salah satu library tersebut adalah scikit-learn.

c. *Reinforcement learning*

Reinforcement learning adalah pendekatan lain untuk pembelajaran mesin, di mana setelah setiap tindakan, agen mendapat umpan balik dalam bentuk hadiah atau hukuman (nilai numerik positif atau negatif).



Gambar 12. *Reinforcement Learning*

Proses learning dimulai dari lingkungan yang memberikan keadaan kepada agen. Kemudian, agen melakukan tindakan berdasarkan tindakan ini, lingkungan akan mengembalikan keadaan dan hadiah kepada agen, di mana hadiah bisa positif, membuat perilaku lebih mungkin di masa depan, atau negatif (yaitu hukuman), membuat perilaku kurang mungkin di masa depan.

Jenis algoritma ini dapat digunakan untuk melatih robot berjalan, misalnya, di mana setiap langkah mengembalikan angka positif (hadiah) dan setiap jatuh angka negatif (hukuman).

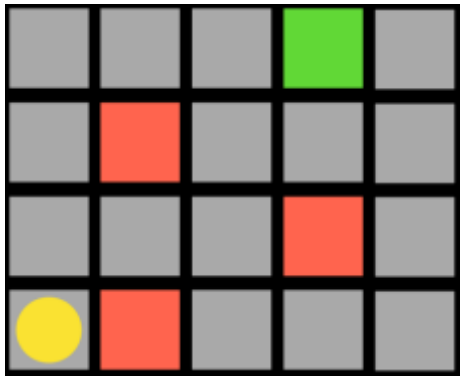
d. *Markov decision Process*

Reinfrocment learning dapat dilihat sebagai proses keputusan Markov, memiliki sifat-sifat berikut:

- Himpunan status S
- Serangkaian tindakan Tindakan (S)
- Model transisi P(s' | s, a)

- Fungsi hadiah R(s, a, s')

Perhatikan gambar berikut ini :



Gambar 13. *Markov Decision Process*

Agennya adalah lingkaran kuning, dan harus sampai ke kotak hijau sambil menghindari kotak merah. Setiap kotak dalam tugas adalah keadaan. Bergerak ke atas, ke bawah, atau ke samping adalah suatu tindakan. Model transisi memberi kita status baru setelah melakukan tindakan, dan fungsi hadiah adalah jenis umpan balik yang didapat agen. Misalnya, jika agen memilih ke kanan, ia akan menginjak kotak merah dan mendapatkan umpan balik negatif. Ini berarti bahwa agen akan belajar bahwa, ketika dalam keadaan berada di kotak kiri bawah, ia harus menghindari ke kanan. Dengan cara ini, agen akan mulai menjelajahi ruang, mempelajari pasangan tindakan-negara mana yang harus dihindari. Algoritma dapat bersifat probabilistik, memilih untuk mengambil tindakan yang berbeda berdasarkan beberapa kemungkinan yang meningkat atau menurun berdasarkan imbalan. Ketika agen mencapai kotak hijau, itu akan mendapatkan hadiah positif, mengetahui bahwa itu menguntungkan untuk mengambil tindakan yang diambilnya di keadaan sebelumnya.

e. *Q-Learning*

Q-Learning adalah salah satu model pembelajaran penguatan, di mana fungsi Q(s, a) menghasilkan perkiraan nilai dari mengambil tindakan a dalam keadaan s.

Model dimulai dengan semua nilai estimasi sama dengan 0 *(Q(s,a) = 0 untuk semua s, a).* Ketika suatu tindakan diambil dan imbalan diterima, fungsi melakukan dua hal:

1) memperkirakan nilai Q(s, a) berdasarkan imbalan saat ini dan imbalan masa depan yang diharapkan
2) memperbarui Q(s, a) menjadi memperhitungkan perkiraan lama dan perkiraan baru. Ini memberi kita algoritme yang mampu meningkatkan pengetahuan masa lalunya tanpa memulai dari awal

B. Unsupervised Learning

Dalam semua kasus seperti dalam *supervised learning* akan memiliki data dengan label yang dapat dipelajari oleh algoritma yang digunakan. Misalnya, ketika ingin mencoba untuk training algoritma untuk mengenali uang kertas palsu, setiap uang kertas memiliki empat variabel dengan nilai yang berbeda (data input) dan apakah itu palsu atau tidak (label). Dalam unsupervised learning hanya data input yang ada dan AI mempelajari pola dalam data ini.

a. Clustering

Clustering adalah *unsupervised learning* yang mengambil data input dan mengaturnya ke dalam kelompok-kelompok sedemikian rupa sehingga objek serupa berakhir di kelompok yang sama. Clustering dapat digunakan, misalnya, dalam penelitian genetika, ketika mencoba menemukan gen yang serupa, atau dalam segmentasi gambar, ketika mendefinisikan bagian-bagian berbeda dari gambar berdasarkan kesamaan antar piksel.

b. k-means Clustering

k-means Clustering adalah algoritma untuk melakukan tugas clustering. Algoritma ini akan memetakan semua titik data dalam ruang, dan kemudian secara acak menempatkan k pusat cluster di ruang (terserah programmer untuk memutuskan berapa banyak; ini adalah keadaan awal yang kita lihat di sebelah kiri). Setiap pusat cluster hanyalah sebuah titik dalam ruang. Kemudian, setiap cluster diberi semua titik yang paling dekat dengan pusatnya daripada ke pusat lainnya (ini adalah gambar tengah). Kemudian, dalam proses iteratif, pusat klaster bergerak ke tengah semua titik ini (keadaan di sebelah kanan), dan kemudian titik-titik tersebut dipindahkan lagi ke klaster yang pusatnya sekarang paling dekat dengan mereka. Ketika, setelah mengulangi proses, setiap titik tetap berada di cluster yang sama seperti sebelumnya, kami telah mencapai keseimbangan dan algoritme berakhir, meninggalkan kami dengan poin yang dibagi di antara cluster.

c. Association

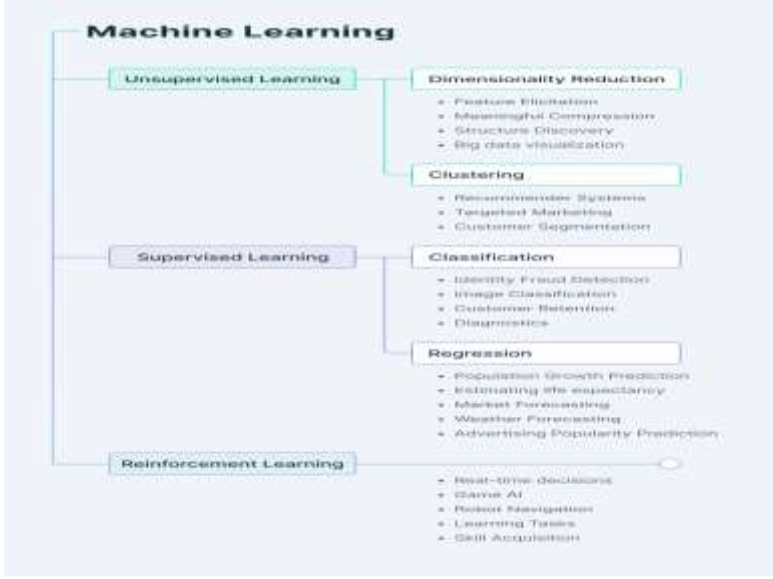
Asosiasi adalah jenis unsupervised learning di mana kita dapat menemukan hubungan dari satu item data ke item data

lainnya. Pada algoritma ini dapat menggunakan ketergantungan tersebut dan memetakannya dengan cara yang menguntungkan misalnya: memahami kebiasaan konsumen mengenai produk kami dapat membantu kami mengembangkan strategi penjualan silang yang lebih baik.

Aturan asosiasi digunakan untuk menemukan probabilitas kemunculan bersama item dalam koleksi. Teknik ini sering digunakan dalam analisis perilaku pelanggan di situs web e-niaga dan platform OTT.

c. Dimensional Reduction

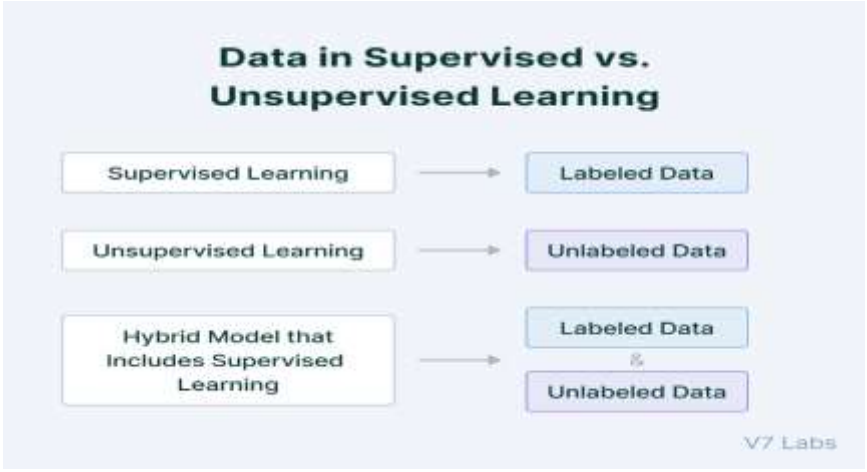
Seperti namanya, algoritma bekerja untuk mengurangi dimensi data. Digunakan untuk ekstraksi fitur. Mengekstrak fitur penting dari kumpulan data adalah aspek penting dari algoritme pembelajaran mesin. Ini membantu mengurangi jumlah variabel acak dalam kumpulan data dengan memfilter fitur yang tidak relevan.



Adapun perbedaan utama Supervised Learning VS Unsupervised Learning adalah jenis data input yang dibutuhkan. Supervised Learning membutuhkan data pelatihan berlabel sementara Unsupervised learning bergantung pada data mentah yang tidak berlabel. Tapi ada lebih banyak perbedaan, dan kita akan melihatnya lebih detail

a. Data

Berikut ini gambar perbedaan Supervised learning dan unsupervised learning dari segi data



b. Tujuan

Tujuan dari Supervised Learning sudah diketahui sebelum pelatihan dimulai.Jenis keluaran yang diharapkan model sudah diketahui; kita hanya perlu memprediksinya untuk data baru yang tidak terlihat. Dengan algoritme pembelajaran tanpa pengawasan, tujuannya adalah untuk mendapatkan wawasan dari sejumlah besar data baru. Tidak ada nilai keluaran tertentu yang kami harapkan untuk diprediksi, yang membuat keseluruhan prosedur pelatihan menjadi lebih kompleks.

c. Aplikasi

Model supervised learning ideal untuk klasifikasi dan regresi dalam kumpulan data berlabel. Deteksi spam, klasifikasi gambar, prakiraan cuaca, prediksi harga adalah beberapa aplikasi yang paling umum.

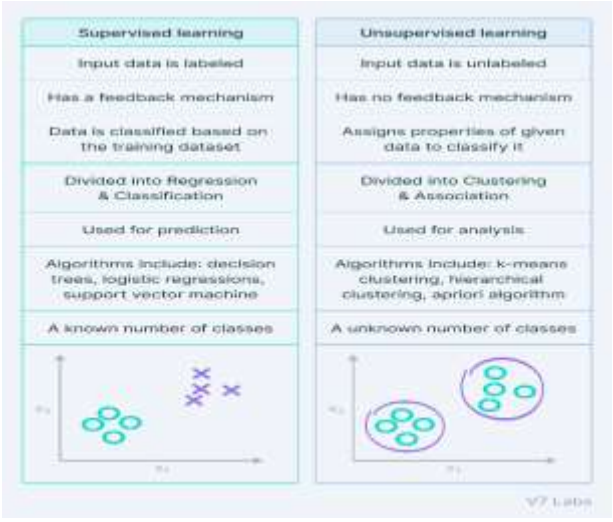
Unsupervised Learning sangat cocok untuk pengelompokan dan asosiasi titik data, digunakan untuk deteksi anomali, prediksi perilaku pelanggan, mesin rekomendasi, penghilangan noise dari kumpulan data, dll.

d. *Complexity*

Supervised Learning relatif kurang kompleks daripada Unsupervised Learning karena outputnya sudah diketahui, membuat prosedur pelatihan jauh lebih mudah.

Dalam unsupervised learning, di sisi lain, kita perlu bekerja dengan kumpulan data besar yang tidak diklasifikasikan dan mengidentifikasi pola tersembunyi dalam data. Output yang dicari tidak diketahui, yang membuat pelatihan lebih sulit.

Lihat tabel perbandingan ini.



10.4 Mamfaat dari *Machine Learning*

Berikut ini beberapa manfaat dari *machine learning* dalam kehidupan sehari hari :

1. Membantu proses penyelesaian masalah bisnis

   Menurut Towards AI , pembelajaran mesin adalah hal yang sangat penting sekarang ini. Machine learning bermanfaat untuk menyelesaikan permasalahan dunia dengan cara yang terukur. Dengan machine learning , seseorang dapat memproses dan menganalisis data yang lebih besar dan rumit dengan waktu yang lebih singkat.

2. Membantu memahami perilaku konsumen

   Salah satu penerapan dari machine learning dalam bidang e-commerce adalah dengan menerapkan machine learning pada suatu sistem sehingga dapat mempelajari perilaku konsumen disebuah aplikasi lalu mengolah data tersebut menjadi layanan personalisasi yang lebih baik.

3. Mewujudkan otomatisasi bisnis

   Automasi sangat membantu meningkatkan efektivitas dan efisiensi operasional perusahan. Salah satunya pada divisi penjualan dan pengelolaan interaksi dengan konsumen.Penggunaan machine learning ternyata terdapat juga pada software *Customer Relationship Management (CRM).* Software ini dapat menggunakan ML untuk menganalisis dan menentukan mana pesan prioritas yang harus dibalas terlebih dahulu oleh tim sales.

4 Mempermudah proses intelegensi bisnis.

   Seorang profesional di bidang *business intelligence* harus menyortir dan memilah mana saja data penting yang berharga bagi perusahaan. Nah, mereka menggunakan perangkat lunak dengan sistem pembelajaran mesin untuk

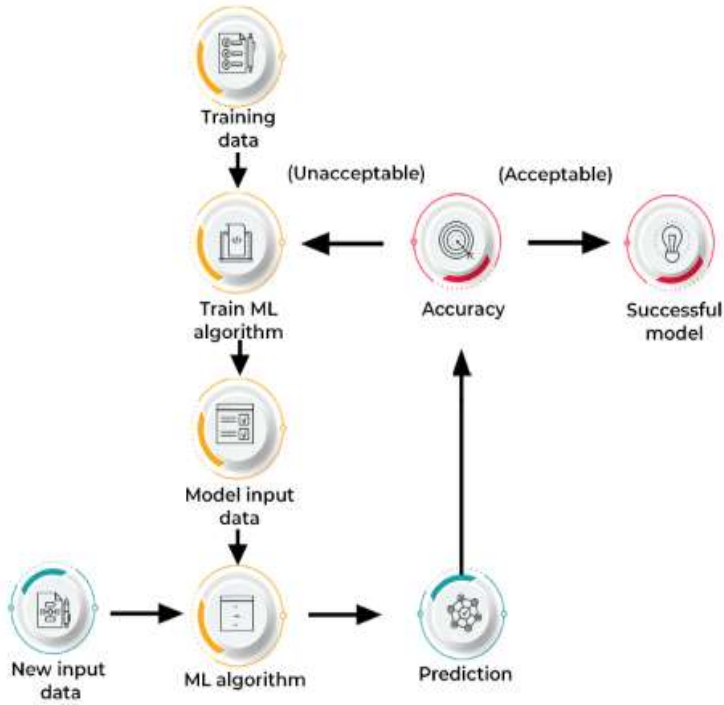
membantu mengidentifikasi titik data, pola data, dan data anomali yang penting.

5. Meningkatkan efektivitas proses SDM

Human Resource Information System atau HRIS merupakan sistem rekrutmen yang sering digunakan oleh para praktisi HR. Sistem ini dapat menggunakan model machine learning untuk dapat mengidentifikasi kandidat dengan potensi paling besar untuk suatu posisi tertentu.

10.5 Bagaimana machine Learning Bekerja

Berikut ini dijabarkan contoh cara kerja dari machine learning *dengan* skenario kasus penggunaan tingkat tinggi. Walaupun bisanya sebuah *machine learning* biasanya bekerja melibatkan banyak faktor, variabel, dan langkah lain.



Cara Kerja Pembelajaran Mesin

10. 6 Penutup

Machine Learning sebagai bagian dari *Artificial intelligence* membutuhkan pengetahuan yang mendasar tentang pola learning itu sendiri diantaranyapembedaan learning dari segi penyajian data, tujuan maupun complexity dari input. Dengan mengetahui tentang konsep dari machine learning sehingga akan lebih mudah untuk memahami tentang artificial intelligence

Bab **11**

Teknologi *Big Data*

11.1 Sejarah *Big Data*

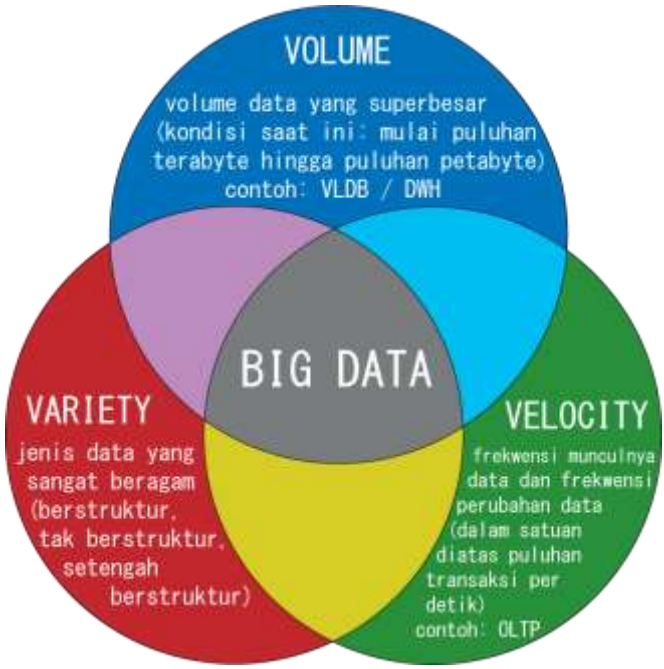
Big Data adalah istilah yang menggambarkan volume data yang besar, baik data yang terstruktur maupun data yang tidak terstruktur. Big Data telah digunakan dalam dunia bisnis. Tidak hanya besar data yang menjadi poin utama tetapi apa yang harus dilakukan organisasi dengan data tersebut. Big Data tentunya mampu untuk dianalisis sebagai penambah wawasan yang mengarah pada pengambilan keputusan dan strategi bisnis yang lebih baik.

(Sumber : https://www.tcgdigital.com/)

Meskipun begitu, Big Data tidak menghiraukan ukuran data, jenis data, atau kecepatan pemrosesannya dapat diartikan sebagai sesuatu yang ridak berharga kecuali jika penggunanya dapat melakukan kegiatan yang mampu menghasilkan value/manfaat bagi organisasi. Meskipun perusahaan/organisasi selalu menjalankan berbagai macam reports dan dashboards

yang berasal dari datawarehouese, namun kebanyakan dari mereka tidak mengeksplorasi secara mendalam isi data tersebut.

Hal ini disebabkan karena alat - alat analytics yang digunakan tersebut terlalu kompleks bagi kebanyakan pengguna pada umumnya dan sebagian sebab lain adalah bahwa tempat penyimpanan data, datanya tidak berisi semua data yang diperlukan oleh para user selaku pengambil keputusan. Tetapi hal ini akan segera berubah dengan cara yang cepat, karena munculnya paradigma *Big Data analytics.*



Gambar 11.2 Karakteristik big data

(Sumber : https://www.kompasiana.com/)

Istilah Big Data masih terbilang baru dan sering disebut sebagai tindakan pengumpulan dan penyimpanan informasi yang besar untuk analisis. Big data atau sering disebut dengan mahadata telah menjadi sering didengar oleh masyarakat

umum untuk sebutan himpunan data dengan jumlah yang sangat besar, tidak terstruktur dan rumit. Sehingga menjadikan big data menjadi sukar ditangani jika hanya menggunakan aplikasi atau software tradisional. Tujuan dari adanya big data yaitu meminimalkan resiko ketidakberhasilan perkiraan dengan mengumpulkan sebanyak mungkin data. Hal ini memungkinkan para pengendali mahadata dapat melakukan hal – hal di masa lalu yang belum dapat dilakukan. Fenomena Big Data, dimulai pada tahun 2000-an ketika seorang analis industri Doug Laney menyampaikan konsep Big Data yang terdiri dari tiga bagian penting, diantaranya:

1. Volume

Volume mengacu pada jumlah data yang akan disimpan. Tabel berikut ini mencantumkan unit kapasitas penyimpanan yang berbeda. Karena bit bersifat biner dan bit merupakan nilai dasar penyimpanan. Sebagai contoh awalan kilo biasanya berarti 1000, tetapi dalam penyimpanan data 1 kb/kilobyte = 1024 byte. Untuk dapat mengelola volume data yang besar, kita memiliki dua opsi untuk menangani beban tambahan (additional load) :

- **Scale Up**: kita dapat menyimpan jumlah sistem yang sama untuk menyimpan dan memproses data, tetapi memigrasikan setiap sistem ke sistem yang lebih besar.
- **Skale Ou**t: kita dapat meningkatkan jumlah sistem, tetapi tidak bermigrasi ke sistem yang lebih besar.

Tabel 11.1 Unit kapasitas (volume big data)

Term	Capacity	Abbreviation
Bit	0 or 1 value	b
Byte	8 bits	B
Kilobyte	1024* bytes	KB
Megabyte	1024 KB	MB
Gigabyte	1024 MB	GB
Terabyte	1024 GB	TB
Petabyte	1024 TB	PB
Exabyte	1024 PB	EB
Zettabyte	1024 EB	ZB
Yottabyte	1024 ZB	YB

Organisasi mengumpulkan data dari berbagai sumber, termasuk transaksi bisnis, media sosial dan informasi dari sensor atau mesin. Aktivitas semacam ini di masa lalu menjadi masalah, namun dengan adanya teknologi baru seperti Hadoop, dapat meredakan masalah ini. Volume atau biasa diartikan dengan jumlah data yang terpenting. Menurut perusahaan oracle, big data akan dapat memproses data tidak terstruktur bervolume tinggi dengan kepadatan rendah. Hal ini dapat berupa data dengan nilai yang tidak dapat diketahui, seperti data Twitter, aliran klik di halaman web atau aplikasi seluler, atau peralatan berkemampuan sensor.

2. Velocity

Velocity atau biasa diartikan dengan kecepatan data, kecepatan aliran data ini harus dapat ditangani dengan secara cepat dan tepat baik melalui hardware maupun software. Teknologi hardware seperti tag RFID (Radio-frequency identification), sensor pintar lainnya pun juga dibutuhkan untuk dapat menangani data yang real-time. Velocity juga merupakan tingkat kecepatan di mana data diterima dan ditindaklanjuti. Biasanya, kecepatan tertinggi aliran data langsung ke memori dibandingkan yang ditulis ke perangkat ketiga.

Gambar 11.3. Variasi data

(Sumber : https://en.wikipedia.org/)

3. Variety

Variasi Data seperti yang telah digambarkan pada Gambar 3, data yang dikumpulkan mempunyai format yang berbeda-beda. Mulai dari yang terstruktur, data numerik dalam database tradisional, data dokumen terstruktur teks, email, video, audio,

transaksi keuangan dan lain-lain. Selain tiga bagian penting tersebut, para peneliti Big Data pun juga telah menambah bagian yang termasuk penting lainnya seperti variabilitas dan kompleksitas. Variety mengacu pada banyaknya jenis data yang tersedia. Tipe data tradisional terstruktur dan cocok tersusun dengan rapi dalam database relasional. Dengan munculnya big data, data datang dalam tipe data baru yang tidak terstruktur. Tipe data tidak terstruktur dan semi terstruktur, seperti teks, audio, dan video, memerlukan prapemrosesan tambahan untuk mendapatkan makna dan mendukung metadata. Lebih jelasnya, data ini mencakup :

- **Data terstruktur,** yang sesuai dengan model data yang telah ditentukan sebelumnya (Data pelanggan, Waktu panggilan , Jenis layanan), dan
- **Data tidak terstruktur** (Rekaman panggilan, sejarah masalah yang terkait dengan panggilan pelanggan).

11.2. Potensi Big Data

Jumlah data yang telah dibuat dan disimpan pada tingkat global hari ini hampir tak terbayangkan jumlahnya. Data tersebut terus tumbuh tanpa henti. Artinya, Big Data memiliki potensi tinggi untuk mengumpulkan wawasan kunci dari informasi bisnis. Sayangnya sampai saat ini, baru sebagian kecil data yang telah dianalisis. Big Data dalam bisnis menjadi strategi yang baik dalam mengolah informasi mentah menjadi keuntungan yang terus mengalir ke organisasi bisnis setiap hari.

Gambar 11.4. Potensi big data

(Sumber : https://phintraco.com/)

Pentingnya Big Data, tidak hanya berputar pada jumlah data yang organisasi miliki, tetapi hal yang penting adalah bagaimana mengolah data internal dan eksternal. Kita dapat mengambil data dari sumber manapun dan menganalisanya untuk menemukan jawaban yang diinginkan dalam bisnis seperti: 1) pengurangan biaya; 2) pengurangan waktu; 3) pengembangan produk baru dan optimalisasi penawaran produk; dan 4) pengambilan keputusan yang cerdas.

11.3. Penerapan Big Data

Dengan adanya 'value proposition', Big Data juga membawa suatu tantangan yang besar bagi perusahaan. Cara-cara tradisional dalam mengambil, menyimpan, dan menganalisa data tidak lagi mampu mengatasi masalah data yang besar secara efektif dan efisien.

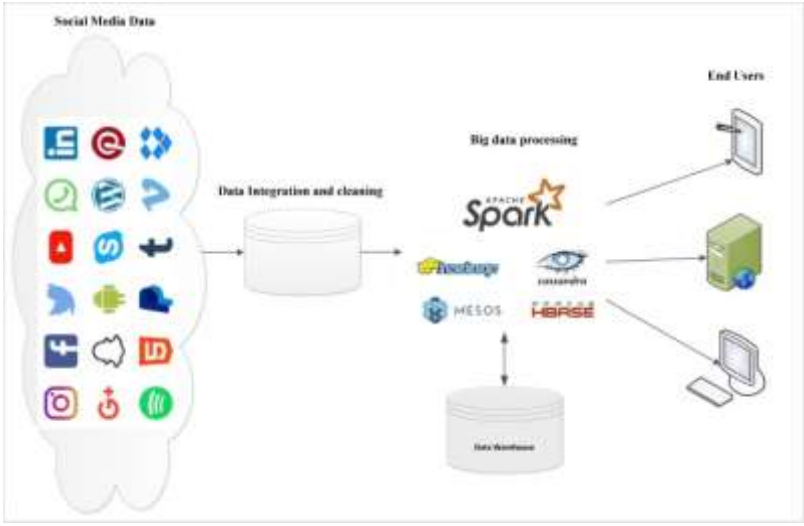
Gambar 11.5. Penerapan Big Data

(Sumber : //www.andrericardo.com.br/)

Karena itu, kegiatan pengembangan teknologi perlu terus dikembangkan supaya dapat mengatasi tantangan datangnya data yang besar. Sebelum melakukan investasi seperti ini, perusahaan pun perlu menganalisa guna membuat beberapa justifikasi. Berikut adalah beberapa pernyataan yang bisa membantu untuk menyoroti situasi tersebut. Bila ada salah satu pernyataan berikut yang benar dengan situasi anda, maka anda pun perlu untuk mempertimbangkan sebelum memulai untuk memulai perjalanan menuju Big Data analytics.

1. Jika didalam kehidupan anda, anda tidak lagi bisa memproses jumlah data yang anda inginkan karena terhalang dengan berbagai keterbatasan yang disebabkan

oleh beberapa platform atau lingkungan yang digunakan.



Gambar 14.6. Pelibatan sumber data dengan platform analytic

(Sumber : https://www.semanticscholar.org/)

2. Ketika anda ingin melibatkan sumber-sumber data baru, misalnya : RFID (Radio-frequency identification), media sosial, Web, data teks, GPS (Global Positioning System) dan lain sebagainya) ke dalam platform analytics seperti yang terlihat pada gambar 6, tetapi anda tidak dapat melakukannya karena tidak cocok dengan skema penyimpanan data yang didefinisikan berdasarkan kolom dan baris.
3. Ketika anda ingin atau mempunyai keperluan untuk dapat mengintegrasikan data secepat mungkin ke dalam analisa yang anda lakukan saat ini.
4. Ketika anda ingin atau sedang bekerja dengan paradigma penyimpanan data 'schema-on-demand' (sebagai kebalikan dari skema yang digunakan dalam RDBMS (*elational database management system*)) karena sifat dasar data yang baru mungkin tidak diketahui, atau

mungkin tidak punya cukup waktu untuk menentukan dan mengembangkan skema seperti itu.

5. Ketika data yang masuk ke perusahaan anda itu, terjadi sangat cepat sehingga platform analytics yang dijalankan secara tradisional tidak mampu lagi untuk menganganinya.

Contoh Penggunaan Big Data

1. **Penggunaan Smartphone**

   Saat ini hampir semua orang dipastikan punya smartphone atau tablet. Smartphone yang anda miliki sebenarnya memiliki jumlah data yang sangat besar. Smartphone mampu menyimpan hasil record telfon dan sms anda. Selain itu, aplikasi-aplikasi handphone anda juga tentunya dapat mengumpulkan data untuk keperluan bisnis anda. Aplikasi GPS seperti Google Maps atau Waze tentunya juga mengumpulkan data-data yang berhubungan dengan lokasi anda

2. **Media Sosial**

   Media sosial tentunya sudah menjadi bagian dari kehidupan manusia sehari-hari. Update photo dan status yang anda upload ke media sosial anda adalah bagian dari data. Setiap harinya ada lebih dari 400 juta tweets yang dikirim ke Twitter dan 72 jam video YouTube diupload setiap menitnya. Tidak hanya itu, dari media sosial, anda juga bisa mendapatkan data tentang kontak kita, hal-hal apa yang sering kita cari dan ikuti di media sosial, dan kebiasaan pengguna media sosial.

3. **Internet**

   Kita semua saat ini telah terhubung dengan internet setiap harinya. Anda juga pasti sering menggunakan Google untuk mencari informasi kan? Nah, data-data dari hasil pencarian

yang anda lakukan tersebut, juga merupakan data yang disimpan Google.

4. **Smart Devices**

   Konsep smart devices sendiri adalah bahwa semua peralatan yang anda miliki di rumah saat ini terhubung satu sama lain dan anda dapat mengaturnya dari satu alat – misalnya smartphone anda. Semua ini merupakan bagian dari teknologi terbaru, biasa anda dengan dengan Internet of Things. Nah, semua data dari smart devices yang anda miliki, seperti misalnya temperatur dan konsumsi daya di rumah anda juga akan dikumpulkan supaya produsen tersebut bisa memperbaiki layanannya dan menawarkan teknologi mutakhir.

5. **Digitalisasi Media**

   Di masa lalu, sebelum maraknya internet seperti saat ini, anda mungkin telah menggunakan CD (*Compact Disk*) dan DVD (*Digital Versatile Disc*) untuk mendengarkan musik atau menonton video. Dengan begitu, anda tidak akan pernah meninggalkan jejak digital. Nah, di era digital seperti sekarang ini, anda mungkin dapat melakukan hal-hal tersebut melalui website ataupun aplikasi streaming seperti Spotify atau Netflix. Tentunya Spotify dan Netflix telah mencatat apa saja yang anda dengarkan ataupun yang anda tonton supaya para produsen memiliki data yang bisa mereka gunakan untuk meningkatkan layanan mereka sesuai dengan jejak rekam digital.

11.4 Penutup

Big data merupakan kumpulan data yang memiliki ukuran yang sangat besar, dimana data – data tersebut terdiri dari data – data yang terstruktur, semi – terstruktur maupun tidak terstruktur. Data – data tersebut seiring dengan berkembangnya zaman dan seiring dengan berjalannya proses

pengumpulan data, jika tidak ada big data di masa tersebut atau di masa yang akan datang, maka kita akan merasa kesulitan dalam menentukan pilihan, menyimpan data – data yang semakin banyak ataupun dalam memilah data besar yang kita punyai. Manfaat big data bisa kita rasakan dari berbagai bidang diantaranya dari bidang bisnis, akademik atau pendidikan, maupun transportasi. Pengumpulan data yang dilakukan big data bisa kita peroleh melalui beberapa program untuk menganalisis aktivitas pengguna media sosial seperti pixlee, crowdfire dan lain sebagainya.

DAFTAR PUSTAKA

Al-Faiz, M. Z., Ibrahim, A. A., & Hadi, S. M. (2019). The effect of Z-Score standardization (normalization) on binary input due the speed of learning in back-propagation neural network. *Iraqi Journal of Information & Communications Technology*, *1*(3), 42–48. https://doi.org/10.31987/ijict.1.3.41

Aprillia, J., 2022. *Apa itu Visualisasi Data? Jenis, Fungsi, dan Toolsnya.* [Online] Available at:

https://www.dewaweb.com/blog/apa-itu-visualisasi-data/

Ardila, Y., Guntoro, Afnarius, S., Santoso, A., Azdy, R., Putra, R., Hasan, N., Sugara, E., Pratama, Y., Fitri, H., Wicaksono, A., & Arnita. (2016). Data Science. In E. Damayanti (Ed.), Penerbit WIDINA (1st ed.). Penerbit WIDINA.

Avendano, Mauricio, and Philipp Hessel. 2015. "The Income Inequality Hypothesis Rejected?" *European Journal of Epidemiology* 30(8):595–98

Bruhn, John G., Betty Chandler, M. Clinton Miller, Stewart Wolf, and Thomas N. Lynn. 1966. "Social Aspects of Coronary Heart Disease in Two Adjacent, Ethnical

Different Communities." *American Journal of Public Health and the Nation's Health* 56(9):1493–1506.

Chamber, J. 2008. Software for Data Analysis. Springer Statistical and Computing. New York: Springer-Verlag.pp1-10

Joe, Yun Jeong Choi, and Yasuyuki Sawada. 2009. "How Is Suicide Different in Japan?" *Japan and the World Economy* 21(2):140–50

Chen, Xiao, Philip B. Ender, Michael Mitchell, and Christine Wells. 2003. *Regression with Stata*. Los Angeles: UCLA

Institute for Digital Research and Education.

D'Agostino, M., & Dardanoni, V. (2009). What's so special about Euclidean distance? A characterization with applications to mobility and spatial voting. *Social Choice and Welfare*, *33*(2), 211–233. https://doi.org/10.1007/s00355-008-0353-5

Data, Falir. 2018. Unvover the R Applications-Why Top Companies are using R Programming, dilihat tanggal 13 Oktober 2022 di https://www.data-flair.training

Egolf, B., J. Lasker, S. Wolf, and L. Potvin. 1992. "The Roseto Effect: A 50-Year Comparison of Mortality Rates." American Journal of Public Health 82(8):1089–92

Emyana Ruth Eritha S, "Implementasi Teknologi Big Data di Lembaga Pemerintahan Indonesia", Jurnal Penelitian Pos dan Informatika,JPPI Vol 6 No. 2, 2016.

Foreman, J. W. (2014). Using Data Science to Transform Information into Insight Data Smart. In *John Wiley & Sons, Inc.* (First edit). John Wiley & Sons, Inc

Goldthorpe, John H. 1997. "Current Issues in Comparative Macrosociology: A Debate on Methodological Issues." *Comparative Social Research* 16:1–26

Goldthorpe, John H. 2010. "Analysing Social Inequality: A Critique of Two Recent Contributions from Economics and Epidemiology." *European Sociological Review* 26(6):731–44

Hill, Catherine. 1992. "Trends in Tobacco Use in Europe." *Journal of the National Cancer Institute Monographs* 12:21–24

Ikeda, Nayu, Eiko Saito, Naoki Kondo, Manami Inoue, Shunya Ikeda, Toshihiko Satoh, Koji Wada, Andrew Stickley, Kota Katanoda, Tetsuya Mizoue, Mitsuhiko Noda, Hiroyasu Iso, Yoshihisa Fujino, Tomotaka Sobue, Shoichiro Tsugane, Mohsen Naghavi, Majid Ezzati, and

Kenji Shibuya. 2011. "What Has Made the Population of Japan Healthy?" *The Lancet* 378(9796):1094–1105.

Ilyas, R. & Pudjiantoro, T. H., 2015. *Visualisasi Data pada Complaint Management System dan Mesin Survey.* Cimahi, Universitas Jenderal Achmad Yani.

J. Grus, *Data Science from Scratch*, 2nd Editio. California: O'Reilly Media, Inc., 201

Jencks, Christopher. 2002. "Does Inequality Matter?" *Daedalus* 131(1):49–6

John F.Quigley,"Consumer Behavior in Digital Markets", Research paper, southern illinois University Carbondale, OpenSIUC,2015.

Joseph, F. 2022. Data Analysis With R. Tersedia URL : https://bookdown.org/jaf005/Data-Analysis-with-R/. dilihat tanggal 24 November 2022

Kennedy, Bruce P., Ichiro Kawachi, and Deborah Prothrow-Stith. 1996. "Income Distribution and Mortality: Cross Sectional Ecological Study of the Robin Hood Index in the United States." *BMJ* 312(7037):1004–1007

Kennedy, Bruce P., Ichiro Kawachi, Roberta Glass, and Deborah Prothrow-Stith. 1998. "Income Distribution, Socioeconomic Status, and Self Rated Health in the United States: Multilevel Analysis." *BMJ* 317(7163):917–21

Kurotani, Kayo, Shamima Akter, Ikuko Kashino, Atsushi Goto, Tetsuya Mizoue, Mitsuhiko Noda, Shizuka Sasazuki, Norie Sawada, and Shoichiro Tsugane, and Japan Public Health Center Based Prospective Study Group. 2016. "Quality of Diet and Mortality among Japanese Men and Women: Japan Public Health Center Based Prospective Study." *BMJ* 352:i1209.

Larose, D. T., & Larose, C. D. (2015). Data Mining and Predictive Analytics. In *John Wiley & Sons, Inc.* (second edi). John Wiley & Sons, Inc

Liao, Tim F. 2014. "Editor's Introduction [to a Symposium on Qualitative Comparative Analysis]." *Sociological Methodology* 44(1):ix–xi

Lillesland, Thomas. M dan Ralph W. Kiefer. 2007. Penginderaan Jauh dan Interpretasi Citra. Yogyakarta. Gadjah Mada University Press

Lindahl-Jacobsen, Rune, Roland Rau, Bernard Jeune, Vladimir Canudas-Romo, Adam Lenart, Kaare Christensen, and James W. Vaupel. 2016. "Rise, Stagnation, and Rise of Danish Women's Life Expectancy." *Proceedings of the National Academy of Sciences* 113(15):4015–20

Ludwig, Jens, Greg J. Duncan, Lisa A. Gennetian, Lawrence F. Katz, Ronald C. Kessler, Jeffrey R. Kling, and Lisa Sanbonmatsu. 2012. "Neighborhood Effects on the Long-Term Well-Being of Low-Income Adults." *Science* 337(6101):1505–10

Madyatmadja, E. D. et al., 2021. Data Visualization of Internet Usage in The Jabodetabek Area. *Infotecch: Journal of Technology Information.*

McCreight, James. 2012. Hands-on R for Climate Data Analysis. NASA Summer Short Course For Earth System Modeling and Supercomputing, dilihat tanggal 10 Oktober 2022 di https://nex.nasa.gov/nex/static/media/other/mccreightHandsOnR.pdf

Melamed, David, Eric Schoon, Ronald L. Breiger, Victor Asal, and R. Karl Rethemeyer. 2012. "Using Organizational Similarity to Identify Statistical Interactions for Improving Situational Awareness of CBRN Activities." Pp. 61–68 in *Social Computing, Behavioral-Cultural*

Modeling and Prediction, Lecture Notes in Computer Science, edited by S. J. Yang, A. M. Greenberg, and M. Endsley. Berlin: Springer.

Melter, R. A. (1987). Some characterizations of city block distance. *Pattern Recognition Letters*, *6*(4), 235–240. https://doi.org/10.1016/0167-8655(87)90082-1

Motohashi, Yutaka. 2012. "Suicide in Japan." *The Lancet* 379(9823):1282–83.

Muharni, S. & Candra, A., 2022. *Modul Visualisasi Data Menggunakan Data Studio,* Malang: Letarsi Nusantara Abadi

Natalia, Natali, V., Harjono, K., WIjaya, C., Putra, R., Hakim, H., Nugroho, P., Karya, G., & Ravi, M. (2020). Pengantar Data Science dan Aplikasinya Bagi Pemula (V. Moertini & M. Adithia (eds.); 1st ed.). UNPAR Press.

Nishom, M. (2019). Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square. *Jurnal Informatika: Jurnal Pengembangan IT*, *4*(1), 20–24.

Nowatzki, Nadine R. 2012. "Wealth Inequality and Health: A Political Economy Perspective." *International Journal of Health Services* 42(3):403–24

OECD (Organisation for Economic Co-operation and Development). 2012. "Income Distribution Data Review— Japan." Paris, France: Organisation for Economic Co-operation and Development.

Omar, T.N., 2017. R Programming by Example. Packt Publishing : Birmingham

Prana, U.G., Adhitya, R.G., 2017. Belajar Bahasa Pemrograman R (Dilengkapi Cara Membuat Aplikasi Olah Data Sederhana dengan R Shiny).USU Press. Medan.

Ramadhani, L., Purnamasari, I., & Amijaya, F. D. T. (2018). Penerapan Metode Complete Linkage dan Metode Hierarchical Clustering Multiscale Bootstrap (Studi Kasus: Kemiskinan Di Kalimantan Timur Tahun 2016). *Eksponensial*, *9*(2016), 1–10. https://fmipa.unmul.ac.id/files/docs/[1] LISDA RAMADHANI 1307015041_Edit.pdf

S. Cooper, "Data Science from Scratch," *CEUR Workshop Proc.*, vol. 1542, pp. 33–36, 2015, [Online]. Available: https://books.google.com/books?id=24kdCAAAQBAJ&pgis=1.

Saddam, H. 2020. 8 Kelebihan dan Kekurangan R: Mengapa Banyak digunakan?, dilihat tanggal 12 Oktober 2022 di https://www.geospasialis.com/kelebihan-kekurangan-r/

Setiawan, A. (2015). Pengantar Teori Probabilitas. In H. Siswanto (Ed.), Tisara Grafika (1st ed.). Tisara Grafika

Sue Yasav," The Impact of Digital Technology on Consumer Purchase Behavior", the journal of Financial Perspectives:Fintech, Vol 3- issue 3, 2015.

T. Barton, *Apply Data Science*. Wiesbaden: Springer Fachmedien Wiesbaden, 2023.

Thomas, Mailund. 2017. Beginning Data Science in R : Data Analysis, Visualization, and Modelling for the Data Scientist. Apress Published : Aarhus, Denmark.

Thomas, Mailund. 2022. Beginning Data Science in R 4 : Data Analysis, Visualization and Modelling for the Data Scientist. Apress Published. Aarhus, Denmark.

Tiobe, Index. 2022. Tiobe Index for October 2022 : The Big 4 Languages Keep Increasing their dominance, dilihat tanggal 13 Oktober 2022 di https://tiobe.com/tiobe-index/

Veikko Halttunsen,"Consumer Behavior in Digital Era",

Academic dissertation, the Faculty of Information Technology of the University of Jyaskyla,2016.

Verzani, J. 2018. SimpleR Using R for Introductory Statistics, dilihat tanggal 12 Oktober 2020 di https://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf

Widodo, B., R.N. Rachmawati. 2013. Pengantar Praktis Pemrograman R untuk Ilmu Komputer. Halaman Moeka Publishing. Jakarta Barat.

Wu, S. 2013. A review on coarse warrantly data and analysis. Reliability Engineering and System, 114. pp.1-11. ISSN 0951-8320.

Yeli, S., 2017. Pemanfaatan Software Open Sources "R" untuk Penelitian Agroklimat. Balai Penelitian Agroklimat dan Hidrologi. Bogor.

PENGANTAR DATA SCIENCE

Data Science merupakan keterampilan yang membutuhkan ilmu komputer, pemrograman, teknologi, dan statistik yang berada di luar rangkaian pelatihan standar bagi peneliti ilmu sosial. Keterampilan ini mencakup teknologi dan teknik seperti memanfaatkan komputasi Cloud, analisis Big Data, pemrosesan Natural Language, pembelajaran tanpa pengawasan (Unsupervised Learning) seperti analisis Cluster, Web Scraping, teknik Fuzzy, Machine Learning, dan lain sebagainya. Data Science dapat membantu peneliti agar dapat bekerja lebih efektif untuk menghasilkan informasi baru yang tepat waktu, menjelajahi kumpulan data yang benar-benar baru dengan cara baru, mengubah pemodelan simulasi, dan lain sebagainya dengan tujuan untuk meningkatkan kuantitas dan kualitas bukti yang diperlukan untuk membuat kebijakan yang lebih baik, memperkuat komunitas, dan meningkatkan kehidupan masyarakat. Seseorang yang memahami Data Science disebut Data Scientist. Seorang Data Scientist tidak harus memahami semua kemampuan yang dibutuhkan karena biasanya Data Scientist bekerja pada tim yang memiliki kemampuan dan keterampilan yang berbeda-beda sehingga dapat saling melengkapi. Secara umum, keterampilan dasar terpenting untuk Data Scientist adalah kemampuan untuk membuat kode dalam setidaknya dua bahasa pemrograman yaitu Python dan R. Keterampilan umum lainnya yang diperlukan oleh seorang Data Scientist adalah keterampilan organisasi yang baik, komunikasi yang jelas, dan kemampuan untuk menguasai konsep dan teknik baru dengan cepat.

TOHAR MEDIA

No Anggota IKAPI : 022/SSL/2019
Workshop : JL. Rappocini Raya Lr.II A No 13 Kota Makassar
Redaksi : JL. Muhktar dg Tompo Kabupaten Gowa
Perumahan Nayla Regency Blok D No 25
Telp. (0411) 8987659 Hp. 085299993635
https://toharmedia.co.id

ISBN 978-623-8148-23-3
9 786238 148233